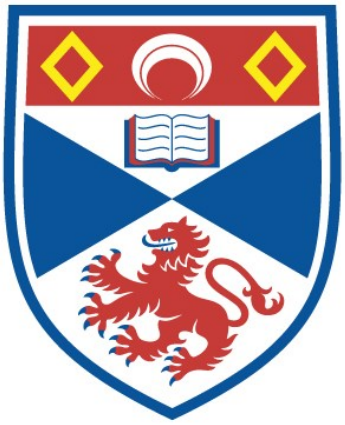
THE DEVELOPMENT AND EVALUATION OF A MICROPROCESSOR-CONTROLLED BIOREACTOR FOR USE IN DEVELOPING COUNTRIES

José Luiz de Lima Filho

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



1987

THE DEVELOPMENT AND EVALUATION OF A MICROPROCESSOR-CONTROLLED BIOREACTOR FOR USE IN DEVELOPING COUNTRIES.

A thesis presented by

José Luiz de Lima Filho

to the University of St. Andrews

in application for the degree of

Doctor of Philosophy.

Biochemistry Department,

The University

St. Andrews.

May,1987

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Th 608

DECLARATION

I hereby declare that this thesis is based on work carried out by me, that the thesis is of my own composition and that no part of it has been presented previously for a higher degree.

The research was conducted in the Department of Biochemistry, University of St. Andrews, under the direction of Dr. W. M. Ledingham.

CERTIFICATE

I hereby certify that José Luiz de Lima Filho has spent nine terms engaged in research work under my direction and that he has fulfilled the conditions of Ordinance General No. 12 of the Resolution of the University Court 1967, No. 1, and that he is qualified to submit the Accompanying thesis for the degree of Doctor of Philosophy.

20.05.87

ACADEMIC RECORD

I graduated with the degree of Doctor of Medicine in 1983 from the Federal University of Pernambuco, Recife, Brazil.

I matriculated as a research student in the Department of Biochemistry, University of St. Andrews, in April 1984.

ACKNOWLEDGEMENT

I should like to express my gratitude to Dr. W. M. Ledingham for his advice during this work, and to my family and friends for their incentive.

I am also grateful to the University of St. Andrews, O.R.S and Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq) for financial support.

INDEX

INDEX

INDEX

INDEX

INDEX

Page

INDEX

Page

Page

Page

Page

INDEX

Page

1. INTRODUCTION

1.1 - Research and Development in Brazil.

1.1.1 - Brazil as part of the International Environment.

"Like America, Brazil is a society of immigrants, unlike America, it is also a society of oligarchs. The immigrants have created one of the world's smartest business cultures. Their entrepreneurial energy has been harnessed, and sometimes suppressed, by one of the non-communist world's most heavy-handed, interventionist and regulation-minded governments" (The Economist, 1987) (1).

In the period 1945-60, Brazil's growth in real Gross Domestic Product (GDP) averaged 7% a year and 10 % in the years of the Brazilian "miracle" (1965-74), and in 1986 the GDP was 11 %. During many of these years most Latin America countries, and in particular the region's other two big economies, those of Argentina and Mexico, were stagnating or contracting. However, with the collapse of the Cruzado Plan (a plan aimed at drastically reducing the countries high inflation) at the end of 1986, the government was befuddle about what to do next, resulting in another in a series of lurches in economic policy that stretches back to the "black years" at the beginning of the 1960s.

In fact the underdevelopment of the Latin-American countries is not an expression of a lack of economical,

political and social developments. It is a consequence of a capitalistic structure, which basically divides the society into two different halves, a society of the rich and a society of the poor (2). The developed countries, in association with multinational corporations, take advantage of the developing countries, particularly in relation to low labour costs and availability of natural resources. The multinational campanies control internal markets and their activities often transcend national interests. For instance, Brazil is the eighth country in western economic ranking (3), but 52 million Brazilians (2/3 of Brazil's population) are living in absolute poverty.

1.1.2 - Historical

During the colonial period, Brazil lagged behind the European countries in development showing no interest in science and technology in contrast to the United States which, as a result, made important developments (2).

In the 19th century, the Portuguese royal family moved to Brazil. As a consequence, many institutions of science and technology were created in Brazil, such as The Medical School in Bahia and in Rio de Janeiro, The National Library and the National Museum, etc. At this time, Brazil was seen as an exporter of natural products, such as minerals and food (2). Although Brazil was a natural resource exporter, science and technology needed to be improved. Consequently, at the beginning of the 20th century, a considerable number of

isolated faculties and institutions of research were developed, mainly in the Rio de Janeiro/Sao Paulo areas, due to their economic and political power. For instance, the Butanta Institute, the Vacinogenico and the Bacteriologico Institutes in Sao Paulo, and most important, the Manguinhos Institute in Rio. This latter institute was created basically because of the idealism of Osvaldo Cruz at the beginning of the 20th century.

Between 1920 and 1930 industrial production in Brazil rose especially in Sao Paulo. However, all of the technology and the majority of the scientists were foreign, there being no interest in the development of a core of Brazilian scientists (4). In the 1950s, Brazil remarked on an ambitious and, by many measures, successful path of industrialisation. This peaked during the country's "golden years" (1968 - 1970) period in which Brazil's talent went to waste as interest in and money for research at universities dwindled (5). This was the period in which Brazil's best scientists went abroad.

Until 1927 Brazil only had three Universities: Parana University (1912), Rio de Janeiro University (1920) and Minas Gerais University (1927). In fact these institutions were a group of several isolated "professional schools". But, only in 1934, with the creation of the University of Sao Paulo (USP) did science and technology improve in Brazil (2).

After the second world war a few institutions were created to stimulate science such as the Brazilian Society for Science Development (SBPC - 1948). In 1951 the first

research council- The National Council for Scientific and Technological Development (CNPq) was created, to provide small grants of a few thousand dollars to individuals for research and for fellowships (2,5). In the same year, the Council to improve the training of university teachers (CAPES) was created (2). Unfortunately, the CNPq and CAPES had their resources cut down when the government took the option to import all technology in the form of "black boxes". In 1967, FINEP (Agency for Financing Studies and Projects) was organised with the same objectives as CNPq and CAPES and grew to became the main source of money for academic science. FINEP, however, split its resources between basic academic research and national industry. Between 1967 and 1984, only 60 per cent of the US$ 1 billion it spent went to universities and other research institutions; most of the rest went as loans to industry (5).

However, the new government of Brazil, the country's first civilian regime in 21 years, has decided that joining the First World requires a heavy investment in science. For the first time, the president has a scientific adviser. Although struggling with a huge foreign debt, the government is funnelling new money into universities and government laboratories for new enterprises such as biotechnology and research in fields such as agriculture that had previously been neglected in favour of industrialisation (5).

1.1.3 - Appropriate Technology and Biotechnology

Appropriate technology can be defined as technology which makes optimum use of available resources (6). Dahalman and Westphal (6) pointed out that because of market failures and superficialities in the creation, diffusion, and choice of technology, there is an important role for government incentives and other interventions in fostering the effective use of technology. The objectives of such policies include inducing the choice of the socially most appropriate foreign techniques; importing technology on the best possible terms; ensuring adequate local participation designed to increase domestic technological capability; and promoting, where appropriate, the use of local rather than foreign resources.

Historically, Brazilian industry has relied on the purchase of foreign technology and on joint ventures with foreign companies. In addition, Brazil's universities have little contact with either industry or the government and conduct little multidisciplinary research (7). However, in 1975, due to the petroleum crises and the adverse relations among universities, industry and government sectors, Brazil developed the National Alcohol Program - PROALCOOL, with the purpose of substitution of gasoline and petrochemicals by alcohol. However, Carioca (8) has pointed out that the technology for distillation employed in the country was introduced in the decade of 1940 and remains unchanged today, in spite of the incredible technological developments introduced in petrochemical fractionation methods (8).

Furthermore, as a result of the use of this outdated technology, a barrel of alcohol costs between US$35 and US$50, and if oil stays below US$15 a barrel, continued alcohol production would mean a theoretical loss of US$ 1 billion a year (9).

In 1982, Gooddrich (7) pointed out a few weaknesses in biotechnology development in Brazil, for instance as Brazil's human resource base trained in advanced biotechnology techniques is limited, the national industrial sector is fairly underdeveloped and has little in-house Research and Development (R&D) capability and little inclination to pursue high-risk ventures. Importation and bureaucratic delays also make it difficult for both public and private laboratories to obtain the necessary R&D equipment and supplies not available on the Brazilian market.

The Biomass Conversion Program has raised public awareness of the important role that biomass can play for the future development of the country. It has also provided an opportunity for scientists and technologists to develop new and appropriate national technologies to reduce the country's dependence on the imported technologies. It should be remarked here that Brazil is already spending a substantial amount of foreign exchange in importing technologies for various sectors of the economy (8).

1.1.4 - Protected Markets.

Many governments adopt protective measures aimed at the development of local technological capabilities. Thus, in 1980, the Brazilian government, in order to protect the emerging Brazilian microcomputer industry, introduced a law "Reserva de mercado" (10), effectively prohibiting importation of foreign computers with very limited exceptions for specialist mainframes. Plans exist to extend this legislation to other 'Strategic' technologies (11). The effect has been a small independence from multinational corporations tempered by a market restricted to out-of-date micros at comparatively high prices and, in all probability, an increase in contraband activities across the border with Paraguay!. A professor at one of Brazil's leading universities, who despite the passing of the military regime prefers to remain nameless, counted himself fortunate in having acquired a foreign computer. "But when it needs any kind of repair, I must take the part to Maimi to get the work done", he said (5).

1.2. COMPUTERS IN FERMENTATION

1.2.1 - Introduction

The first generation of computers appeared in the 1950s using valves as the main component. The second generation based on transistor technology started in the 60s. The third generation, where the Z-80 Central processing Unit was included (12), was developed in the 70s using integrated circuit technology at that time minicomputers being the most important. Subsequently, appeared the fourth generation characterized by the development of the microcomputers. The birth of the fifth generation with the creation of supercomputers was announced by the Japanese in 1983.

Unlike the development of the minicomputer, closely associated with the American space effort of the 1960's, or the mainframe, which grew out of wartime weapons and code cracking research, microcomputer development took place outside the mainstream of computer science. Table 1.2.1a shows significant events in microcomputer evolution, from 1971. But only in the past five years has computer control made significant impact on the fermentation industry (13). The decrease in cost and improved reliability of computer hardware in recent years have made the use of computers even more attractive (14).

Computer technology has improved so fast that, for example, a typical microcomputer today has the computing power comparable to that possessed by a minicomputer in the

early 1970s and to that possessed by a mainframe computer in the early 1960s. In the past years, there has been an amazing proliferation of computer applications in the petroleum and chemical industries. This has occurred because computers have proved useful in handling large masses of data with high operational speed (13). For the fermentation industry, economic pressure and competition make the use of computers stil more critical. Although the number of environmental sensors has increased, the need for better and more efficient methods of data acquisition, analysis and documentation remain a pressing problem. Table 1.2.1b shows some of the different areas of computer application in fermentation.

Table 1.2.1b - Some areas of computer application in fermentation.

Instrumentation
Data acquisition
Data analysis
Mathematical modelling
Control
Optimisation

Table 1.2.1a - Significant events in microcomputer evolution.

Year	Event
1971	INTEL develops the first microprocessor - The 4004.
1973	National Semiconductor Corporation launches a printed circuit board with processor and memory chips for control applications.
1974	MIT produces the first commercially available microcomputer in kit form, the Atari.
1977	Apple, Commodore and Tandy bring out ready-built microcomputers.
1981	IBM, Hewlett-Packard and Xerox, established computer companies, bring out microcomputers, confirming the importance of personal computing.
1983	The first 32-bit microprocessor appears putting the theoretical processing power of a micro on a par with that of a mainframe.

1.2.2 - Historical

The Chemical process industry began utilising direct digital control (DDC) in the late 1950s (14). But only in the 1960s was the idea of using computers to control fermentation first proposed by Fuld (15).

Armiger and Hamprey (16) suggested that, as a result industrial secrecy, it is not clear which companies were the first to utilise computers in fermentation processes. However, there are some publications dealing with industrial applications. In 1966, the Ajinomoto Company in Japan utilised a YODIC-500 computer for direct digital control of their glutamic acid fermentation, and Dista Products utilised an ARCH 102 computer for direct digital control in a new fermentation plant. In 1969, Glaxo Laboratories, Ltd., converted an existing plant from analog to direct digital control.

Computer-controlled fermentation was first described in 1969 by Grayson (235) and Yamashita (63), who reported control of sequential operations in fermentation such as start-up, sterilisation and shutdown (13).

In 1971, a philosophy of data acquisition, analysis and computer control of fermentation process was presented by Nyiri (17), and in 1972, he reviewed an article about off-line applications of computers in biochemical engineering in data analysis, simulation, optimal trajectory calculation, and parameter estimation for kinetic model construction (17).

In 1977, Dobry and Jost (18) reviewed papers from the previous five years, about computers to control fermentation in the industry (19). In 1977, Zebriski et. al. (20) pointed out two main problems in this area. First, the practical limitations of control in real-time and, secondly, availability of sensors. In 1979, Hampel (21) reviewed control procedures giving special emphasis to microcomputers application.

Many of the variable values needed to analyse a fermentation process are directly measurable. For those variables which are not measurable, the concept of 'gateway sensors' has played an important role. By this technique, the values of certain biologically significant parameters may be calculated from data from a combination of available sensors. The computer, in this case, is very useful in making the necessary calculations, thus providing a 'gateway' to calculate unmeasurable quantities (for example, biomass concentration). Measurable quantities which are frequently used for these purposes include the concentration of oxygen and carbon dioxide in the exit gas (16).

In 1982, Rolf and Lim (14) and Hatch (22) reviewed the existing hardware and software technology available for the computer control of fermentation process, showing the concept of interfacing techniques, data logging and documentation, low-level control, back-up and error detection and programming techniques.

In 1984, Buckland (23) reviewed the application of computer process control in industrial pilot plant, emphasising the application of hierarchial systems within a network.

In 1986, Merill and Bauer (24) developed an integrated microprocessor based fermenter control system which can be used in both batch and continuous process. It consists of a two-level hierarchical system: 1 - A microprocessor-based local controller and; 2 - A minicomputer. An 8-bit microprocessor was chosen because its capacity provides the demands of local control and because interfacing to external devices is cheaper. The software was organized around a multitasking real-time operating system, with application programs and a modular database with one module for each operating variable.

The reviews above described the complexity of computer control fermentation, and how expensive computer control can be. Since 1979, the utilisation of microcomputers in 'stand alone' mode in laboratory-scale fermentations or coupled into a network with minicomputers and mainframe computers has improved especially because of the increased processing capacity of microcomputers and decreased price of their components.

1.2.3 - Computer Hardware and Software.

System requirements inevitable vary with the particular application. Each microbial process is different and the complexity and the objectives of the application may dictate certain choices in the design criteria. This section discusses the basic components of a microcomputer system normally used to control fermentation. Microcomputers have been used to control fermentations since 1979 as shown in table 1.2.3 (13).

Table 1.2.3 - Partial list of microcomputers used to control fermentation.

System	References	
Single Microprocessor board to control a single fermentation vessel.	Jefferies et. al.	(25)
Apple II coupled with off-gas analyzers.	Forrest et. al.	(26)
ST-6001 Process Controller to control dissolved Oxygen.	Yano et. al.	(27)
DEC-MINC 11 coupled with MASS Spectrometer for monitoring fermentation liquids and gases.	Punger et. al.	(28)
Commodore CBM 3020 coupled fermentation	Bayer et. al.	(29)
Apple II+/ISAAC41A coupled with mass flow controllers to control gases	Mathers and CorK	(30)
Hitachi H685B02-1 used to control dissolved oxygen, carbon dioxide and nutrient feed rates.	Shimizu et. al.	(31)
Apple II+/ISAACC91A for controlling batch and continuous fermentations	Titus et.al.	(32)
Hewlett-Packard HP-85/HPIB interface to control	Podruzny & Van den Berg	(33)

anaerobic methane fermentation		
Commodore CBM 64 coupled fermentation used to investigate wastewater treatment systems.	Horan et. al.	(34)
Microcomputer automated HPLC monitoring of fermentation system introduced.	Dinwoodie and Mehnert	(35)
Apple IIe for fermentation control.	Strohl et. al.	(36)
Hewlett-Packard HP-1000 for controlling continuous cultivation of yeast.	Chen et. al.	(37)

Microcomputer control of fermentations offers several advantages to researchers such as:

1 - Extensive data-logging of parameter values which can be carried out at time intervals prescribed by the operator;
2 - Increased accuracy and reliability by averaging of many data points using algorithm-directed rejection of noisy and false signals and periodic internal recalibration;
3 - Experimental variable manipulation in the absence of the operator;
4 - Implementation of variable control strategies to manipulate a process along an optimum trajectory;
5 - Fermentation control via calculated variables in real-time.

Microcomputers can be used as stand-alone systems to monitor the progress of fermentations, collecting and storing information about a fermentation for later analysis, and they can perform modest on-line calculations that describe the progress of the fermentation.

In many cases microcomputers, because of their limited memory, are used as low level controllers in a hierarchical computer control process involving minicomputers or mainframes.

1.2.3.1 - Hardware

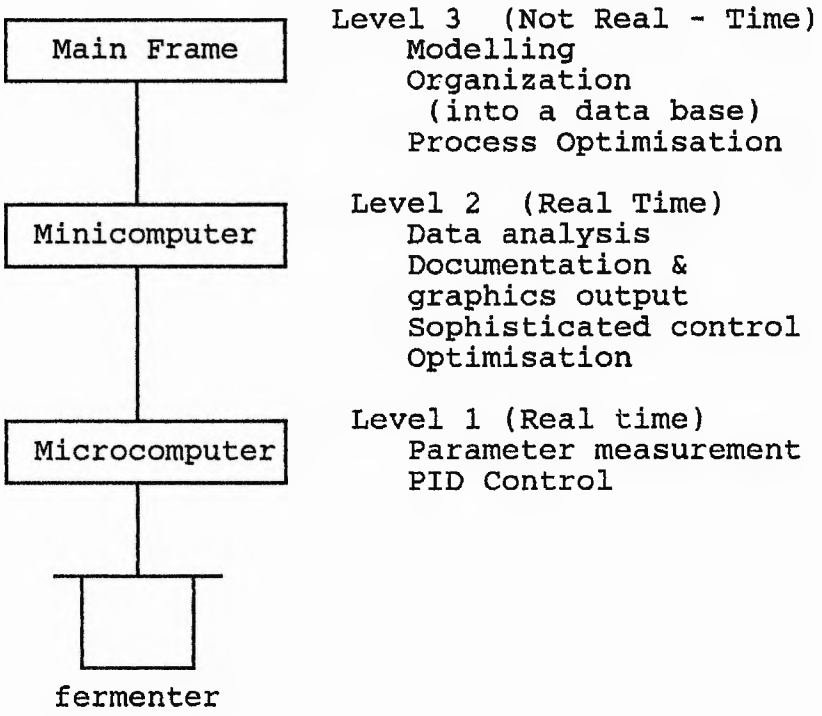
Wang and Stephanopoulos (13) have described two kinds of computer-coupled systems in existence today. The first type is more pilot-plant or production-oriented where the computers are connected in a network with several fermenters working at the same time. The second category is more laboratory-scale oriented with, usually, only one fermenter being controlled in which all of the data acquisition, data analysis, control and optimisation functions may be carried out simultaneously. The application of microprocessors in the evaluation of intelligent interfaces and instrument systems has also been suggested by Yasuda (38).

Rolf and Lim (14) suggested that selection of the computer system depends on such factors such as the control requirements of the fermentation process, the number of units to be interfaced, the extent of processing functions needed other than control, and availability of existing computer hardware. Therefore, a large number of applications recently have used hierarchical computer systems.

Figure 1.2.3.1.1 shows a diagram of a hierarchical system (23, 39, 40, 41) described by Wang and Stephanopoulos (13), which represents a three level structure used for a large system. The microcomputer, as the low-level controller, has high priority on-line measurement and simple proportional-integral-derivation (PID) control tasks. The second control level, usually a minicomputer, is frequently

reserved for on-line data analysis and sophisticated control algorithms aimed at optimisation. The third level, represented by a mainframe, is where process modelling is carried out.

Figure 1.2.3.1.1 - Hierarchical computer system used for a large system



A hierarchical system used in an industrial pilot plant, which has four levels of function, has been described by Lundell (42) as follows:

1 - Production planning;

2 - Production control;

3 - Unit process control;

4 - Local process control.

The system is modular in structure enabling a straightforward expansion. It can be used for Direct Digital Control (DDC), a technology described by Beaverstock and Trearchis (43).

Hennigan et. al. (47) pointed out some advantages of the utilisation of hierarchical systems, for instance, their similarity to the use of subroutines in computer programming in the sense that one can change hardware and/or software in one computer without affecting the assigned routines in other computers; and in the event of failure in the smaller computer the rest of the system is not affected.

The usual PID control may be substituted by conventional analog controllers with set points supplied by a computer, in which case DDC degenerates into digital set point control (SPC). Table 1.2.3.1.1 shows same advantages of DDC.

Table 1.2.3.1.1 - Advantages of Direct Digital Control

Versatility
No limitation in control techniques
Control algorithms are modified easily
Controller parameters are adjusted easily
Decreased hardware requirements
No need to design, construct, calibrate
or service analog controllers
One control routine serves for several loops
Improved response
Better regulation
Smoother start-up and shut-down
Bumpless transfer in switching with manual systems
Easy measurement of manipulated variables
Automatic logging of control actions

In SPC the computer is involved only in decisions on the setpoint for the low-level control loops. Therefore, in the event of computer failure, the low-level control loops remain operational. The main disadvantage of SPC is that the

type of control algorithms for hardware controllers is limited (14). However, in DDC control, the computer must be available constantly and cannot be diverted to service other functions while DDC programs are running (14, 44).

Multicomputer systems are not necessary for every application in computer-controlled fermentation, but their reduction in hardware cost and the advances in support technology now makes the use of microcomputers justifiable in many cases.

1.2.3.2 - Interfacing

Interfacing required between a computer and the measurement and control instrumentation associated with a fermenter differs little from that required for other applications. The connection between the real physical world of the fermenter and the abstract world of the computer is through interfacing, and an adequate method must be employed to make the computer work as the user desires. There are commonly four kinds of interfacing, as follows:

1 - Transducers --» Computer (data for logging and control action).

2 - Computer --» Computer (Intercomputer communication).

3 - Computer --» Actuator (e.g., valves, switch, pumps, etc., for control action).

4 - Operator --» Computer (Interaction with human «-- world, supervisionary control).

Analog to digital (A/D) and digital to analog (D/A) converters are routinely used in interfacing in the fermentation fields in any other field. Analog-to-digital converters translate analog measurements to digital or binary codes for use in data processing, transmission, and control

systems. Digital-to-analog converters are used in transforming digital data (from computations) back to the analog form for control, display or analog processing (12).

1.2.3.3 - Software

Process-control systems software employs many approaches that are common to other software writing tasks. However, in the real time environment of process-controlled fermentation, a high speed of execution and operation are desirable (45, 46, 47). Process control software may be divided into 3 levels:

1 - Initialisation software (e.g. sterilisation, setting-up the fermentation conditions, temperature, pH etc.).

2 - Control software (e.g. to maintain fermentation conditions constant).

3 - Supervisory software (e.g. to calculate parameters, for instance, specific growth rates, etc. Carry out optimisation and modelling from data collected previously).

Wang and Stephanopoulos (13) pointed out that, for process control software, programs are written to compare measured values to parameter set points and thus generate

controller outputs. A sophisticated control scheme allows optimisation and, hence, calculation of best operating conditions and internal generation of parameter set points.

Nyiri (17) divided a program package into 2 major phases, namely:

1 - The preparatory phase - Where two basic functions are performed:

a) Calculations of tolerance values for the process variables which are introduced through the operator's console;
b) Sterilisation.

2 - The executive phase - Containing the program routines for data logging, data analysis and process control.

In 1983, Meiners and Rapmundt (48) pointed out that the most important part of applications software has to be developed by the scientist, therefore ease of use/adaptation of this software is an important feature. They used four types of programs: Main schedule tasks, low priority on-line tasks, local closed circuit loops and off-line programs. It should be realised that straightforward calculations, such as applying mass balances to calculate biomass concentration, represent a small fraction of total software (less than 1%).

Process Control Software has been defined by Spector

(45) as a collection of separate but cooperating tasks. A task, as defined by him, is an independent sequence of computer instructions that can call specific data. Multiple tasks can be carried out on multiple processors, with each processor executing a single task. It is desirable that process control software supports an available high-level language such as Fortran, Basic or PL/M (25). Several systems, however, also use low level languages such as Assembler or a special process-control language such as IMPAC (33) or Pearl (Process and Experiment Automation Real-time Language) (49).

Rolf and Lim (14) pointed out that three basic requirements for control software are:

1 - That it is developed as a real-time priority-based operating system (with multi-programming capabilities).

2 - It is modular (allowing incorporation of a new application without extensive re-organisation).

3 - It is "user-friendly" permitting easy use and modification by the operator.

1.3 - INSTRUMENTATION

The monitoring and controlling process depends on available instrumentation (14). This is one area where the needs of fermentation technology exceed those of the conventional chemical industry (13), essentially because the dynamic processes of living organisms are far more complex than chemical industry processes.

Much work has been directed towards increasing the availability of different types of sensors for on-line measurement.

Measurable variables can be divided into two categories:

1 - Environmental variables { temperature, pH, dissolved oxygen, agitation speed, aeration rate.

2 - Physiological variables { Products of metabolism - Ethanol, penicillin, etc.
State of metabolism - specific growth rate, etc.

Many of the variables needed to analyse a fermentation process are directly measurable (14) using appropriate transducers.

1.3.1 - Transducers.

A Transducer is a device for determining the value of a physical or biological variable (12). Transducers should have a rapid response from the point of view of both process modelling and control implementation and be sterilisable. Table 1.3 shows the features of an ideal transducer for monitoring parameters in fermentation.

Table 1.3 - Ideal characteristics of a transducer to be used in fermentation analysis.

Rapid response
Low cost
Sterilisable
Sensitive
durability
Non-destructive assay
Biologically inert
Cleanable *in situ*
On-line measurement

In the following sections some characteristics of transducers normally used in fermentation process control will be described.

1.3.1.1 - Temperature Transducers

Temperature measurement in fermentation can be carried out using one of the following sensors: Resistance temperature detectors, thermistors, transistors, integrated circuit (I.C.) transducers, thermocouples, thermal pyrometer (12). Thermistors are most widely used in fermentation because of their high sensitivity and the absence of the cold junction requirement, however thermistors have the disadvantage that output is highly nonlinear, but this can be resolved via the use of software (13).

1.3.1.2 - pH transducers

pH is one of the most widely monitored variables in fermentation. Steam sterilisable pH probes are commercially available in the form of a combined unit for the H^+ half-cell and the reference half-cell.

1.3.1.3 - Measurement of gases

1.3.1.3.1 - Dissolved Oxygen transducers

Oxygen electrodes can be classified as either potentiometric (galvanic) or amperimetric (polarographic or Clark type), depending on whether a current is forced through them or not. It should be noted that these electrodes actually measure the activity, or equivalent partial pressure, of the dissolved oxygen (oxygen tension), and not the concentration (13).

In 1964, the first autoclavable dissolved oxygen electrode was described by Johnson. In 1984, Bungay (86) developed a microelectrode to measure oxygen in gas and liquid phase. In the same year, Reviv and Ben-Yaakov (87) developed an algorithm for predicting BOD (Biochemical Oxygen Demand) from dissolved oxygen (DO) data which may be implemented on a microcomputer for fast on-line determination of BOD (61).

1.3.1.3.2 - Dissolved Carbon Dioxide transducers

Reliable steam-sterilisable sensors to measure dissolved carbon dioxide were unavailable until very recently for the reason that dissolved CO_2 concentration is assumed to be in equilibrium with gaseous CO_2 concentration in fermenter exit gases - This latter being measurable with available CO_2 gas analysers (13).

The commercially available electrode which measures HCO^{-3} ions requires the conversion of HCO^{-3} to CO_2 gas and operates over a very strict pH range. However, Shoda and Ishikawa (88) have described a simple laboratory-constructed steam-sterilisable electrode made by adapting a common pH electrode.

1.3.1.3.3 - Measurement of gaseous O_2 and CO_2.

The continuous measurement of gaseous oxygen concentration with a paramagnetic O_2 analyzer and of carbon

dioxide concentration with an infrared CO_2 analyzer has become quite routine in fermentation processes (13). Oxygen and carbon dioxide concentration data subsequently can be used for biomass determination as described elsewhere (89).

1.3.1.4 - Biomass measurement

Biomass concentration is an important parameter in various stages of the fermentation monitoring and control process. However on-line biomass determinations have not been yet accomplished in a totally satisfactory manner.

Wang and Stephanopoulos (13) and Harris and Kell (90) divided methods to measure biomass into the following categories: Chemical, Physical, Optical, Thermal, Mechanical and Manual, depending on the principles behind the measurement.

1.3.1.4.1 - Chemical methods for biomass measurements.

Chemical methods require that a sample be removed from the fermenter and are thus inappropriate for real-time monitoring. They include chemiluminescence (91, 92), measuring the ATP concentration (93, 94, 95), NADH concentration under certain conditions (96, 97, 98, 99, 100) and concentration of nucleic acids (101), proteins (102) and phospholipid (103). King and White (89) correlated the concentration of muramic acid, which is found only in

bacterial cell walls, including those of blue green bacteria, with biomass.

Warren (104) and Trinel et. al. (105) proposed an enzyme-based method using a colorimetric β-Galactosidase assay and another based on the glutamic acid decarboxylase activity, to measure the concentration of faecal coliforms in water (106). In 1983, Lloyd (114) used a mass spectrometry technique to correlate biomass concentration with a variety of gases in aqueous and gas phase, such as oxygen and carbon dioxide.

Norris and Swain (108) and Adams (109) determined biomass concentration using microscopic methods with an appropriate staining and counting chamber. Jones (110) has improved this method of measuring biomass using epiflourescence microscopy which utilises acridine orange, a stain with a high affinity for nucleic acid.

1.3.1.4.2 - Physical methods for biomass measurements

In principle, physical methods are better for adapting to continuous, in situ, monitoring of biomass in fermenters.

Dry weight is often taken as the definitive measure of biomass, but because samples have to be removed from the bulk culture and are subsequently discarded, it is not a suitable technique for the continuous monitoring of fermenters, but may, like viable counts, provide a one-off spot check (90). Because of the hygroscopic nature of microbial cells, dried

cells absorb moisture during weighing, which can result in an error of approximately 0.6% of the total dry weight.

The photometric system has been classified by Harris and Kell (90) into two categories: nephelometric and turbidimetric. The nephelometric one, which consists of the measurement of scattered light, has been used since the beginning of the 1960s (111,112) to measure the concentration of microorganisms in suspension.

In 1983 Wilson and Harvey (113) applied the light scattering principle in the form of a two-beam laser velocimeter which measures the passage of cells between the two laser beams.

Transmitted light or optical density (Turbimetry) (114) measurement of a microbial suspension depends on the geometry of the instrument, the wavelength of light, the relative refractive index and the cuvette path length. It is also a function of microbial size and shape which may alter with growth phase (115). It provides a relatively quick and convenient way of estimating cell concentrations in a fermenter. Highly automated turbimeters, designed for clinical analysis, are available. Hancher et. al. (116) have suggested that this method is unsatisfactory for used in fermentation processes for the following 4 reasons:

1 - Gas bubbles present in the medium may interfere.
2 - Other (non-microbial) particulate material may be present.
3 - The microorganisms are normally grown at

concentrations outside the limits of linearity between optical density and cell concentration;

4 - Wall growth may occur especially on the photosensor.

Aldridge et. al.(117), Heslop (118) and Gibson (119), have used the scanning of the entire sample-containing cuvettes for elimination of the effect of gas bubbles. Ohashi et. al. (120) overcame this problem by designing a submersible colorimetric probe that permits easy and continuous measurement of the optical density whilst removing the air bubbles from the sample. However, the problem of cell growth on the surface of the probe does not yet seem to have been solved generally (90). Pirt (121) showed that the adhesion of biomass can be temporarily prevented by application of silicon on the surface of the vessel or prevented permanently by using teflon membrane.

For reliable and reproducible data, both in nephelometry and turbimetry, a dry weight relantionship must be established.

1.3.1.4.3 - Other Photometric Techniques for biomass measurements.

Kull and Cuatrecasas (122) applied a photometric method to estimate the number of viable cells using a neutral red stain .

A semicontinuous automated biomass probe using a

filtration method to monitor the growth of Penicillium chrysogenum for penicillin production was developed by Nestaes and Wang (123).

Since last century it has been known that changes in the electrical impedance of microbial cultures is associated with microbial growth (124, 125). The measured change in impedance is a function of the type and number of microorganisms, the medium in which they are growing, the frequency of the applied ac signal, the surface properties and geometry of the measuring electrodes, the surface to volume ratio, the temperature and the interelectrode distance (90).

An electrical counting system which monitors the effect of microorganisms on an electric field as the microorganisms traverse the electrical field, has been developed by Grover et. al. (127, 128).

In 1974, Ackland et. al. (190) developed a simple electrical method which allows the direct enumeration of microorganisms in various substrates without dilution. It requires a cell assembly which is filled with substrate, and growth is detected by monitoring the voltage produced across the electrodes.

In 1979, Wilkins (129, 130) used an electrical system for the evalution of microbial population, based on the detection of gases produced using a membrane-covered electrode.

1.3.1.5 - Other parameter measurements used in fermentations.

A foam detector which works through a conductance impedance measurement was developed by Gualandi (131).

On-line specific-ions probes for a series of mineral ions are commercially available for Na^+, K^+, Mg^{2+} and Ca^{2+} concentrations. Other ions present in the fermenter such as PO_4^{3+}, SO_4^{2-}, Cl^-, NH_4^+ and NO_3 can also be monitored. Unfortunately, these probes can be operated only between certain limited pH ranges and are not steam sterilisable (13).

The last ten years has seen an increase in sensors based on immobilised enzyme technology. Because of their specificity for a particular chemical compound, a very wide range of biochemicals can be assayed selectively by employing suitable enzymes. Guilbault (132), Bakers and Soners (191) and Iannielle and Jesperson (133) have reviewed this subject. The use of immobilised whole microbial cells or organelles as sensors has also been investigated quite intensively (13).

In 1981, Suzuki and Karube (134) used an oxygen electrode, in conjunction with immobilised aerobic microorganisms, for the measurement of glucose, acetic acid, alcohols, ammonia, nitrate, and biochemical oxygen demand (BOD), based on the determination of the rate of microbial respiration. Mass spectrometry has also been applied for the monitoring of low molecular weight compounds in fermenter exit gas (135, 136, 137).

One of the most obvious changes which occurs during the growth of all microorganisms is the production of heat (90). Heat production in principle can be used to quantify microbial population (134, 138, 139). In 1963, Calvet and Prat (140) developed a fast and ultrasensitive microcalorimeter for biological applications. But its direct use in a typical small scale fermenter has been considerably restricted because of the low rate of heat production.

Wang and Stephanopoulos (13) classified calorimeters into two groups: 1 - Adiabatic calorimeters where the cell is thermally insulated so that the amount of heat exchange between the measuring cell and the surrounding environment depends on the temperature of the system; 2 - Heat-conduction calorimeters in which the heat generated inside a measuring cell is conducted through the wall to the surrounding constant-temperature heat sink. The rate of heat flux through the wall is generally proportional to the difference in the temperature between the cell and the heat sink. In 1982, Cheung et. al. (192) developed a data-acquisition and reduction procedure for a precise adiabatic calorimeter at cryogenic temperatures in strong magnetic fields.

A gas chromatograph with a computer-controlled sampler injector to measure several compounds from the fermenter broth culture was used by Comberbach and Bu'lock (141). In 1982 Wang and Stephanopoulos (13) suggested that this principle can also be applied to high performance liquid

chromatography (HPLC) procedures.

Continuous determination of ethanol from exit gas has been measured by Bach et. al. (142) using a semiconductor sensor (smoke detector) during anaerobic cultivation of yeasts.

Hatch et. al. (143) and Fazel-Madglessi et. al. (144) have used a microflourometry technique for measuring DNA, RNA and specific proteins by single-file flow of cells through a beam of light.

A bioelectrochemical system (146) to measure biomass has been developed by Ramsay et. al. (145). The system consists of a reaction cell in which potassium ferricyanide is reduced by the microbial biomass and then reoxidised by the electrode. The resulting current is proportional to microbial concentration with a response time of approximately 3 minutes.

1.4 - CONTROL

The application of process controlling in industry started at beginning of this century (see table 1.4) (50). Devices used to control parameters which are easy to measure, such as temperature, have been used effectively since the 1930s. However, the idea of applying computers to control a fermentation process was proposed only in the early 1960s.

Table 1.4 - Evolution of technologies in Industrial Process Control

Period	Type of Control
1900-1915	Manual with pneumatic control (valves)
1915-1930	Proportional pneumatic control
1930-1945	Control by feedback using a proportional integral differential (PID) approach.
1945-1960	Electronic Controls (e.g. relays). Application of computers to control processes.
1960-1987	Direct digital control (DDC). Application of computerised hierarchical systems.

It is a generally accepted idea that, to obtain more efficient control over fermentation processes, real-time computer control must be operative. Humprey (51) suggested that to achieve meaningful control of fermentation processes

it is necessary to:

1 - To carry out fermentation research in a fully monitored fermenter environment;

2 - To correlate observations from the fermentation with existing knowledge of cellular metabolic response;

3 - To refine the environmental control system according to the results of such correlation;

4 - To reproduce the optimal environmental conditions through continuous computer control.

Onken and Wiland (52) pointed out three types of control actions, proportional (P), integral (I) and differential (D). Depending on the time behaviour of the controlled variable, P, PI, PD, or PID controllers are used. In proportional control (P), the controller produces an output signal which is proportional to the difference between the set-point and the sensor signal. Hence, with proportional control, the greater the deviation of the sensor signal from its set-point, the stronger is the corrective action. Integral (I) action gives a control output signal which is proportional to the time integral of the error, whereas differential (D) action gives an output which is proportional to the derivative of the error.

Rolf and Lim (14) classified computer control into two areas:

1 - Low-level control in which a digital computer is used to replace conventional analogue controllers;

2 - "Modern" control in which the process is optimised using a digital computer.

They suggested a simple model where it should be clear how the techniques can be extended to more complex models. They suggested an unstructured model on the basis of mass balance equations for the cells, substrates and product, as follows:

$$(d/dt)\ (VX) = FiXi - FX + \mu XV \qquad VX(0) = VoXo \qquad (1)$$

$$(d/dt)\ (VS) = FiSi - FS - \sigma XV \qquad VS(0) = VoSo \qquad (2)$$

$$(d/dt)\ (VP) = FiPi - FP + \pi XV \qquad VP(0) = VoPo \qquad (3)$$

$$(d/dt)\ (V) = Fi - F \qquad V(0) = Vo \qquad (4)$$

V = Bioreactor volume
X = Cell concentration
S = Substrate concentration
P = Product concentration
π = Specific production rate
Fi = Inlet flow rate
F = Output flow rate
μ = Specific growth rate
σ = Specific substrate consumption rate

The subscript i denotes those variables associated with the inlet flow stream. Equations (1) to (4) represent mass balance equations for a batch, fed-batch or continuous bioreactor. For a batch reactor Fi=F=0, while for a fed-batch

reactor F=Xi=Pi=0. In the continuous reactor, Fi=F and Xi=Pi=0.

Computer control has been divided by Specter (45) into three different levels:

1 - Data acquisition;

2 - Control of parameters based on information supplied by the transducers;

3 - Optimisation.

Wang and Stephanopoulos (13) have listed four main reasons to utilise computers on-line with bioreactors:

1 - Computers, via sensors, allow the receipt of frequent updates on parameter values (unlike a manual sampling procedure).

2 - Use of parameter data acquired for comparision with set-points and subsequent control by feedback.

3 - Use of experience gained with small scale computer-controlled bioreactors in the design of control for large scale processes.

4 - The possibility of digital set-point control (set-points being computer determined to optimise performance).

Waite and Gray (53) performed an on-line optimisation study to maximise biomass productivity in a steady-state continuous fermenter. The optimal biomass concentration was monitored in real time with continuous optical density measurement, and the dilution rate was adjusted accordingly to keep the cell concentration at the desired value so as to obtain maximum productivity.

In 1983, Mou and Cooney (54) used an on-line growth monitoring and control system. Carbon-balancing equations were used to calculate the cell concentration and specific growth rate in the penicillin production phase. In the same year, Comberbach and Bu'Lock (55) applied a computer-controlled gas chromatograph to monitor the progress of ethanol production from both aerobic and anaerobic continuous fermentations.

Märkl et. al. (56) have used an on-line microcomputer to monitor an anaerobic methane digestion process with the aid of a mathematical model. They have used this system to look at the fluctuations in the gas production caused by a change in substrate (acetic acid) concentration. In 1984, Wu et. al. (57) suggested a method of on-line optimal control for fed-batch culture of baker's yeast production. The control policy was to maintain the specific growth rate at a given value. The optimal feed rate of a fed-batch culture is then obtained by minimizing the performance index.

In 1985, Rolf and Lim (58) and Miskiewicz and

Miskiewick (59) suggested an adaptive on-line optimisation method that utilises dynamic modelling to maximise the cellular productivity of a continuous baker's yeast culture. In 1986, a novel method of growth monitoring of aerobic microorganisms in a laboratory scale fermenter was developed by Walczak (60), using an oxygen sensor coupled with a microcomputer.

In the early years of computer control in biochemical engineering, sensors were inefficient necessitating algorithms to filter noise, etc.

In the area of biomass production, computers have permitted a substantial improvement especially in feedback control to regulate the nutrient feed rate ("feed-on-demand" control), and the indirect determination of biomass using 'gateway sensors' (see section 1.2.2).

At present, controls are rarely carried beyond manipulation of the extracellular environmental conditions of temperature, pH, dissolved oxygen and substrate concentration. However, more imaginative multivariable interaction controls are expected to appear in the future.

1.4.1 - Macroscopic Material/Energy Balances

In 1970, Fredrickson et. al. (61) applied the term "balanced growth" to a method which implies that as the amount of biomass in a certain culture varies with time, the composition inside the cell remains unchanged. Consequently, all changes in the amount of macromolecules are proportional to the change of the overall amount of biomass (62).

The principal macroscopic material balance and energy balances have long been known in chemical engineering. However, Yamashita et. al. (63) were the first to attempt to measure indirectly biomass and growth rate on-line using macroscopic balance in fermentation. There the total rate of biomass growth R(in g/cm^3) was determined by:

$$R = (Y/V)\ (Fin - Fout)$$

Where, V was the volume of the fermenter, Y the yield of biomass with respect to the component under consideration (g biomass/ g component), and Fin and Fout were the rates (in g/s) at which a component that is consumed during fermentation is supplied and removed from the fermenter, respectively.

However, the question is, how to achieve effective control in a fermentation system ?. Herein lies one of the major bottlenecks in the development of computer-aided fermentation: The sensors for direct monitoring of biomass, product (other than biomass) concentrations, and substrate concentration are not available for use in commercial-scale

fermentations (63). However, in 1977 Cooney et. al. (64) suggested a method that combines the use of commercially available sensors in order to determine indirectly the biomass concentration at any time during the fermentation. It is based on the concept of material balances. The basic feature of the method is to represent the conversion of substrate to cell mass and metabolic product by a chemical reaction as follows.

Equation 1

carbon
energy + oxygen + ammonium -» cells + water + carbon + Product

Equation 2

$aCxHyO_2 + bO_2 + cNH_3 \text{ -» } BC_\mu H_\beta O_\alpha N_\delta + dH_2O + eCO_2 + fC_{\mu'}H_{\beta'}O_{\alpha'}N_{s'}$

In equation 2 a, b, c, d, e and f are moles of the carbon source and cell mass, μ and α are constants. For the moment, all the chemical formulae are assumed known and constant, although there are reports that the chemical composition of cells may be affected by drastic changes in growth rates and by the nature and composition of the medium. Herein lies the disadvantage of this method. There are six unknown stoichiometric coefficients for the reaction, and the stoichiometric coefficients for biomass can be normalised to one. The principle of elemental balances gives four equations, one for each of the elements C, H, O, and N, as follows:

C: $ax = \mu + e + \mu'f$
H: $ya + 3c = \beta + 2d + \beta'f$
O: $2a + 2b = \alpha + d + 2e + \alpha'f$
N: $C = \delta + \delta'f$

A quasi steady-state approximation applied to the conservation equations for O_2 and CO_2 gives a fifth equation needed to solve for the six unknowns:

$$R = \frac{YO_2}{V} = OUR = \frac{YCO_2}{V} = CER$$

Where R is the total rate of growth, YO_2 and YCO_2 are the yields of biomass with respect to oxygen and carbon dioxide, respectively, and V is the volume of the culture reactor (in litres). From the measurement of the flow rates and the concentrations of O_2 and CO_2 in the inlet and outlet gas streams, the OUR and CER can be readily calculated on-line (14). The OUR and CER are also related to the stoichiometric coefficients in the equation below:

$$\frac{CER}{OUR} = RQ \text{ (respiratory quotient)}$$

From these stoichiometric coefficients, all the yield relationships can be obtained. Thus, this material balance method does not require the assumption of constant cellular yield coefficients. As a matter of fact, no kinetic models are used at all.

Zarbriskie et al (65) also described a second method

for indirect estimation of biomass concentration and growth rate through on-line material balance. It is based on the material balance of only one chemical component. Mathematically, the relation of formation/utilisation of the chemical component under consideration to biomass growth, biomass and/or substrate concentration can be continuously estimated.

1.5 - Modelling

Mathematical models are routinely used in chemical engineering in various phases of operations from preliminary equipment design to the final specification of operating conditions and optimisation. In contrast, existing mathematical models are rarely considered adequate by investigators in the biochemical engineering field.

In 1949, Monod (66) described a mathematical model of the relationship between microbial cell growth and substrate utilisation, whereas 1959 Leudeking and Pirt (67) reported a mathematical relationship between growth and product formation.

In 1972, Nyiri (17), described two types of models related to microbial growth, namely:

1 - Unstructured models in which a uniformly distributed cell mass is considered along with the relationship between biomass and product formation;

2 - Structured models where the cell structure is also considered In this model the total nitrogen content is differentiated (DNA, RNA, protein); in addition there is a possibility of considering enzyme activities and metabolic pathways.

The use of models for the simulation of the dynamic

behaviour of the system modelled to determine optimal environmental conditions has been discussed by Nyiry (17).

In 1986 Bolle et. al.(68) described a dynamic model of a continuous working up-flow anaerobic sludge blanket (UASB). The model is able to predict the various observable and nonobservable or difficult to observe state variables.

Modelling should somehow reflect the present state of knowledge. It should never proceed beyond such a point where the increased complexity no longer contributes to understanding. It is almost always much easier to fit the same experiment data to a sophisticated model containing many dynamic equations and a multitude of adjustible parameters than to a simple model containing only a handful of these dynamic equations and a comparatively small number of parameters.

History has shown that simplicity shall prevail, and herein lies the difficulty in modelling, i.e., how to explain intricate responses with simple lucid models.

1.5.1 - Types of Modelling

Monod (69) proposed that the growth rate of an organism might be related to substrate concentration in a manner analagous to that between velocity and substrate concentration in enzyme kinetics. This seemed reasonable since one could postulate that there might be a single limiting enzyme reaction that controlled the growth rate of the cell for a given type of substrate. The Monod model can be written as:

$$\frac{dx}{dt} = \mu X \text{ , where } \quad \mu = \frac{\mu max.S}{S + Ks}$$

X = Biomass concentration

S = Substrate concentration

Ks = Saturation concentration

μ = Specific growth rate

μmax = Maximum specific growth rate

Many variations to the Monod model have been proposed and are summarised by Fredrickson and Tsuchiya (70).

Garfinkel (71) has classified modelling into two classes: Discrete event models, where a succession of separate events is considered and continuous models, where system behaviour is followed as a function of time.

Models have been divided by Kassen (72) into two categories, as follows: phenomenological (unstructured) and mechanistic (or structured):

1 - Phenomenological models, sometimes also called formal, empirical, or black box models, are mainly used to describe the overall observed microbial response.

2 - Mechanistic models structure the cell into many components in order to provide explanations for the observed phenomenon.

However, to achieve an exact mathematical model from the phenomenological viewpoint is more difficult than using the mechanistic model. However, Takamatsu et. al. (73) have suggested that some approximate mathematical (phenomenological) models are usually necessary to design and control a fermentation plant.

Wang and Stephanopoulos (13) classified models into:

1 - Statistical or deterministic, depending on whether randon effects are considered;

2 - Segregated or unsegregated, depending on whether the distribution of products in a cell population is differentiated;

3 - Lumped or distributed, depending on whether spatial variations in bioreactor conditions exist.

The characteristics of a proposed control strategy can be studied through the use of mathematical analysis and the

application of control theories. Wang and Stephanopoulos (13) are concerned that many revisions of the model may be necessary and each time a new control strategy may be formulated.

Mathematical models have been used in different fermentation areas, helping to understand the physiology of microorganisms. In 1979, Boutlon (74) used mathematical models to formulate control strategies for wine production, for example to control the fermentation heat production rate in order to avoid overload on the cooling system.

In 1981, Barford (75) developed a mathematical model for the aerobic growth of Saccharomyces cerevisiae, based on a saturated respiratory capacity, in batch and continuous culture. In the same year, a simulation of the integration between the internal energy metabolism and the cell cycle of Saccharomyces cerevisiae, was carried out by Hall and Barford (76).

Dynamic modelling of continuous culture systems for process control and optimisation has been developed by O'Neil and Lyberatos (77), data for such modelling being obtained from the outcome of feedback control designed to destabilise a steady state.

In 1986, Lim et. al. (78) developed an efficient algorithm for fed-batch fermentation process using four or less mass balance equations. This algorithm calculates the optimal substrate feed rate profiles, it is straightforward

and simple for various fed-batch cultures.

A mathematical model within the "cybernetic framework" for the diauxic growth of Klebsiella pneumoniae on a mixture of D-glucose and D-xylose has been developed (81, 79, 80). The cybernetic view of the cell presumes the existence of certain structured functional parts, such as adaptive ones, permanent ones, and regulator ones.

1.6 - OPTIMISATION

Optimisation is the ultimate objective of applying computer (83, 85, 84) control to fermentation processes. However, delays in the realisation of this goal have been caused by:

1 - Availability of appropriate sensors (e.g. biochemical sensors);
2 - Availability of mathematical models to describe microbial culture behaviour;
3 - Lack of resources.

San and Stephanoupolos (189) classified the area of fermenter optimisation into two major areas:

1 - Static optimisation - used generally to optimise performance of continuous microbial cultures.

2 - Dynamic optimisation - applied to the more complex situation of batch and fed-batch operations.

Optimisation does not involve only the fermentation process, it also covers operations such as filtration, extraction, drying, solvent recovery, waste treatment, etc., which follow the growth phase.

1.7 - BIOLOGICAL SYSTEM USED FOR EVALUATION OF COMPUTER-CONTROLLED FERMENTER

1.7.1 - Introduction.

The choice of a strain of Saccharomyces cerevisiae as a biological system to evaluate the computer-controlled fermenter (developed in this work) was related to a biotechnologically important - Brazilian industry - the sugar-alcohol industry, which is a very important part of the economic activities of Brazil, especially of the North East.

Saccharomyces cerevisiae is an eukaryote that, like bacteria, can be grown and analysed genetically without difficulty. In addition, considerable effort has been given to the manipulation and modification of the genome by classical techniques, mainly mutation and hybridization. This knowledge has provided the essential background for gene cloning and host-vector development (147).

Among the environmental conditions that commonly influences the properties of microbial cells in nature, the concentration of essential nutrients is of particular importance (148). Nitrogen is an essential nutrient and, depending on the properties of particular organisms, may be supplied in the form of an organic nitrogen compound, ammonia, nitrate, or molecular nitrogen. Many organisms are able to utilize ammonia, which is mainly incorporated into glutamate or glutamine, and these compounds in turn serve as the main precursors for the synthesis of other cellular

organic nitrogen compounds (147, 148). The engineered strain used in this project was transformed by a plasmid, which carries the glutamate dehydrogenase (NADP-GDH) gene conferring a 10 fold increase in activity (149, 150).

1.7.2 - Ammonia assimilation.

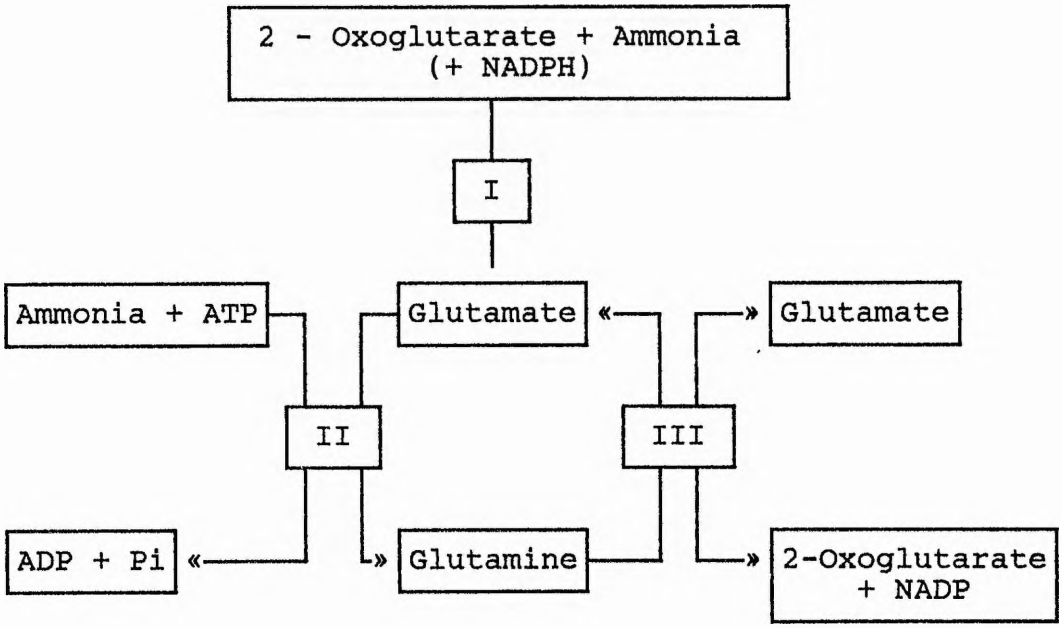
Cooper (151) pointed out that in Saccharomyces cerevisiae the routes of nitrogen catabolism may be conveniently divided on the basis of their end products. Several systems, such as those degrading allantoin, urea or asparagine, generate ammonia as the final product. In these cases, NADP-GDH is required to convert ammonia to glutamate, the predominant nitrogen donor in many biosynthetic reactions.

Ammonia can be assimilated into microorganisms by incorporation into glutamine or glutamate. Glutamine synthetase (GS), an ATP-requiring enzyme, catalyses the first of these reactions, and, in many microorganims, serves together with glutamate synthetase (GOGAT i.e. glutamine 2-oxoglutarate amino transferase) as the main ammonia-scavenging pathway which is strongly induced under nitrogen limitation (152, 153, 154). This pathway has been described by Meers et. al. (155) and Tempest et. al. (156) where it has been shown that glutamate is synthesised by a two-step process that involves first the amidation of endogenous glutamate to glutamine and then the reductive transfer of the glutamine amide-nitrogen to the 2-position of 2-oxoglutarate, as shown in the figure 1.7.2a. The latter

step involves GOGAT. NADP-GDH catalyses the second reaction, and is usually regarded as a high capacity/low affinity enzyme, mainly functional under unlimited ammonia supply (157, 158, 159). In organisms having both aminating pathways, GS is drastically decreased upon an increase in ammonia supply, and amination proceeds via NADP-GDH (154).

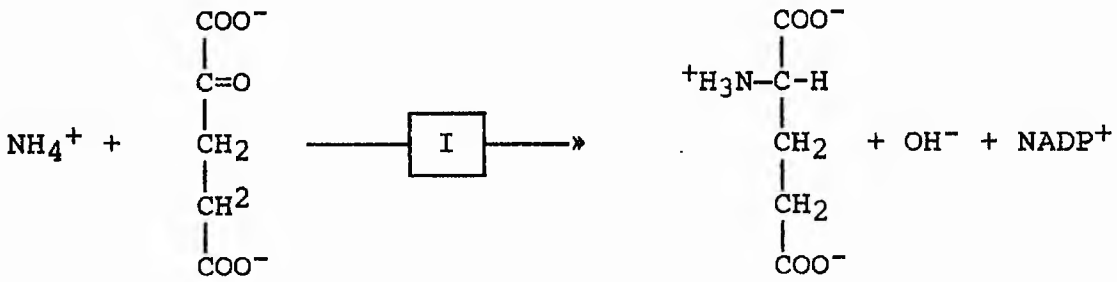
Figure 1.7.2 - Synthesis of glutamate from ammonia in

Saccharomyces cerevisiae



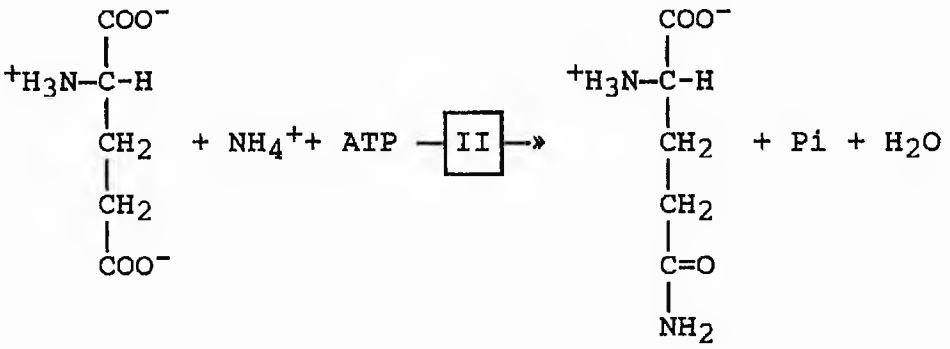
I -» Glutamate Dehydrogenase:

Ammonia + 2-oxoglutarate + NADPH —» Glutamate + H_2O + $NADP^+$



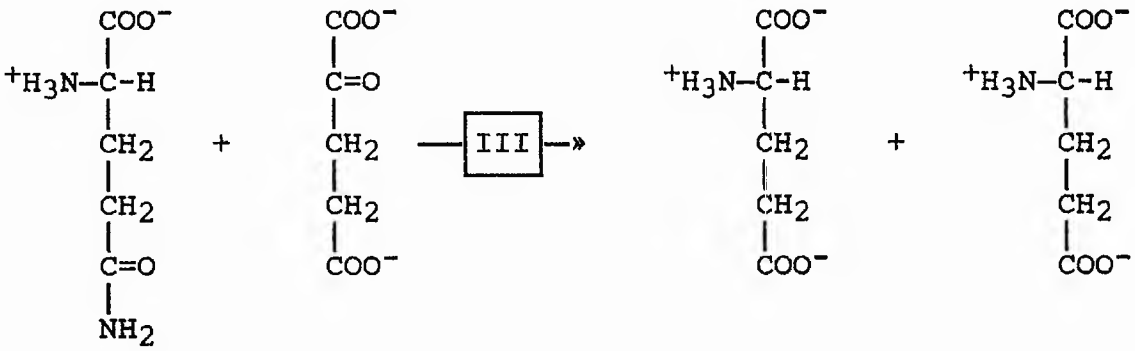
[II] -» Glutamine synthetase:

Ammonia + Glutamate + ATP —» Glutamine + ADP + Pi + H_2O



[III] -» Glutamine amide-2-oxoglutarate aminotransferase (oxidoreductase, NADP): GOGAT (Glutamate synthase)

Glutamine + 2-Oxoglutarate + NADPH —» 2 Glutamate + $NADP^+$



Reactions [II] + [III] :

Ammonia + 2-oxoglutarate + ATP + NADPH —» Glutamate + ADP + Pi + $NADP^+$ + H_2O

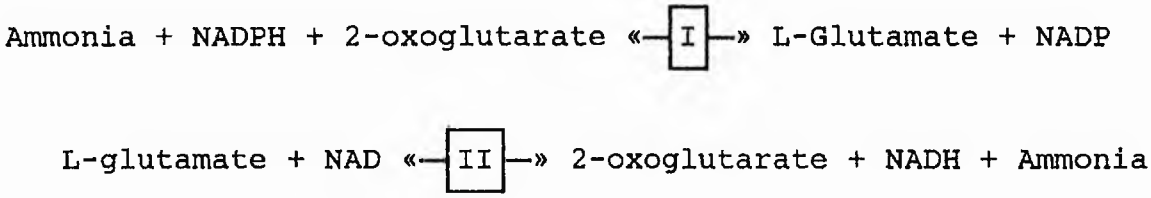
Roon et. al. (160) suggested that the irreversibility and high affinity for ammonia of both synthases of this pathway can function in a very efficient manner when the cellular levels of free ammonia are low. They also pointed out that variations in the specific activity of GOGAT, which occur in growing yeast cultures in response to change in the nitrogen source, generally parallel changes observed with NADP-GDH (anabolic), and also that the maximum level of activity is more than 10-fold higher than the maximal level of GOGAT activity. They also suggested that yeast GOGAT serves in a capacity which is auxiliary to the NADP-GDH, perhaps as a means for directly converting the amide group of glutamine to α-amino nitrogen without contributing to the pool of free ammonia.

Glutamine synthetase from Saccharomyces cerevisiae can be regulated by at least three different mechanisms (161), namely repression of its synthesis by ammonia, glutamine-mediated inactivation, and proteolysis under nitrogen and carbon starvation. The bacterial GOGAT allows the organism not only to assimilate ammonia but also to use it at a much lower concentration than is possible when the bacterium possesses only NADP-GDH. This is possible because of the expenditure of one molecule of ATP. In the absence of glutamate synthase from a strain with normal NADP-GDH activity, growth is not modified even under conditions in which ammonia is limiting.

Brown (162) suggested that, in bacteria, the first enzyme of ammonia assimilation under most conditions is

GOGAT. In contrast, in many fungi the first enzyme involved is NADP-GDH. Saccharomyces cerevisae synthesises 2 glutamate dehydrogenases (163, 164, 165) (figure 1.7.2). One of these enzymes is specific for NAD and the other utilises only NADP.

Figure 1.7.2 - Interconversion of glutamate and ammonia by the 2 GDHs.



Kohlhaw et. al. (166) described high levels of the NAD-specific GDH when cells were grown on glutamate and low levels when ammonia was provided as nitrogen source. This prompted the conclusion that NAD-specific GDH is a catabolic enzyme. The loss of activity when cells were grown on glutamate plus ammonia was interpreted as an indication that the catabolic enzyme was subject to ammonia repression. Growth conditions that yield high expression of NADP-GDH cause low level expression of NAD-GDH, and vice versa (151). On this basis, it has been suggested that the NAD-dependent enzyme functions in a catabolic capacity catalyzing the breakdown of glutamate to ammonia and 2-oxoglutarate (165).

Moye et. al. (150) pointed out that the NAD-specific glutamate dehydrogenase is controlled by reversible

phosphorylation/dephosphorylation (167) while NADP-glutamate dehydrogenase is inactivated and rapidly degraded in response to severe nitrogen or carbon limitation (168).

Hernandez et. al. (169) and Roon and Even (165), observed a maximal activity of NADP-GDH when ammonia is the sole nitrogen source. However, Bogonez et. al. (153) have shown that the activity of NADP-GDH of Saccharomyces cerevisiae decreased under conditions in which intracellular ammonia concentration increases, suggesting that this decrease is brought about by repression of synthesis.

There is some evidence that 2-oxoglutarate, or some other intermediate of the tricarboxilic acid cycle, is a positive modulator of NADP-GDH induction (170). Gonzalez et. al.(170) also observed a decrease in NADP-GDH when the specific growth rate was around $0.15h^{-1}$. Furthermore, Bogonez et. al. (153) demonstrated an increase of 2-fold in NADP-GDH activity on cells grown on 10mM ammonium sulphate compared to cells grown on 20mM glutamate, irrespective of the carbon source. The slow growth on glutamate was associated with a corresponding increase in internal glutamate levels. However, the concentration of ammonia within the cell was not modified, with the exception of the increase in lactate-grown cells which has been shown to result from the rise in pH that accompanies the growth of yeast on weak acids (171).

Ammonia transport in yeast is mediated by two active transport systems that operate within a narrow pH range (5.5

- 7.5) with maximal activity at pH 6.5 (172). However, Bogonez et. al.(171) pointed out that at higher pH values ammonia uptake is essentially a diffusion process.

1.7.3 - Yeast strain genetic engineering.

1.7.3.1 - Introduction

Recombinant DNA research shows great promise in furthering understanding of yeast biology by making possible the analysis and manipulation of yeast genes not only in the test tube but also in vivo (173).

Yeasts are eukaryotic microbes whose cells, despite their small size, bear complex cytoplasmic organisation similar to those of higher eukaryotes and serve as useful tools for molecular understanding of eukaryotes (174).

The genome size of the well-studied budding yeast Saccharomyces cerevisiae is as small as 1.0×10^{10} daltons per haploid cell (175), with 80% of the total genome resident in chromosomal DNA. The detailed genetic map is constructed on 17 chromosomes on which more than 300 genes are sited (176).

Ephurssi et. al. (177) were among the first to show the existence of genes in the cytoplasm, which showed that the inheritance of a mitochondria petite mutation of Saccharomyces cerevisiae follows the non-Mendelian law. In 1979, Bevan and Mitchel (178) were the second to show a cytoplasmic genome in Saccharomyces cerevisiae, the killer yeast that carries double-stranded (ds) RNA species, encapsulated in the virus-like particles, called ScV.

Yeast cells also harbour a 2μ DNA of unknown function in 25 to 100 copies per cell, amounting to about 4% of the total DNA. Gunge (174) and Gubbins et. al. (179) demonstrated the existence of a 2μ DNA in a mixture of two isomer forms distinguishable from each other by relative orientation of their L and S regions, the entire sequence consists of 6318 bp, which are separated by the identical inverted repeats of 599 bp into two unique sequences of 2774 bp (L region) and 2346 bp (S region) (174). The 2 μ DNA replicates only once at an early stage of S phase in the cell cycle (180), and the copy number per cell appears to be genetically controlled (181). One of the most striking features of 2μ circles is that they are present in nearly all strains of Saccharomyces cervisiae, including laboratory strains, brewing and other strains of commercial importance (182).

There now exist simple and general methods for isolating and amplifying virtually any yeast gene. Although these methods generally require an intermediate step in Escherichia coli (173), which offers high efficiency of transformation, good amplification and has a powerful hybridization screening system (173).

1.7.3.2 - Vectors

Four classes of plasmid yeast vector systems useful to the molecular biologist have been described by Old and Primrose (183), as follows in the table 1.8 below.

Table 1.8 - Some properties of different yeast vectors.

Vector	Transformation frequency per µg DNA	Loss in non-selective medium
YIp	1 - 10	Much less than 1% per generation
YEp	10^3 - 10^5	1% per generation
YRp	10^2 - 10^3	Much greater than 1% per generation but chromosomal integration is possible
YCp	10^2 - 10^3	Less than 1% per generation

The four kinds have been named after the way in which they are maintained in yeast after transformation (173) such as:

YIp (Yeast integrating plasmid) - plasmids that must be

maintained in yeast after transformation.

YEp (Yeast episomal plasmid)- use a fragment of the 2μ plasmid for maintenance.

YRp (Yeast replicating plasmid) - contains an autonomously replicating sequence.

YCp (Yeast centromere plasmid) - contains a functional centromere.

The method of yeast transformation is based on the incorporation of exogeneous DNA into yeast protoplast in the presence of polyethylene glycol and the regeneration of protoplasts into normal cells in hypertonic selective media. Gunge (174) classified the transforming DNAs into two classes:

1 - Integrating plasmids - Plasmids constructed by insertion of yeast genomic DNA (lacking a replicator) into an Escherichia coli plasmid.

2 - Autonomously replicating plasmids - Plasmids converted to autonomously replicating plasmids capable of efficient transformation, by incorporation of yeast replicator sequences from 2μ plasmid, chromosomal or mitochondrial DNAs.

1.7.3.3 - Plasmid stability in yeast genetic engineering.

Imanaka and Aiba (184) have defined the stability of

recombinant plasmids as the ability of transformed cells to maintain plasmid unchanged during their growth, manifesting their phenotypic characteristics. They also suggested that stability of a plasmid may be affected by several factors such as:

1 - Growth rate;
2 - Genetic characteristics of host cells and the plasmid (copy number, relaxed or stringent type of replication control, physiological consequences of gene expression on the plasmid),
3 - Environmental stress imposed on host cells, etc.

In addition Caulcott (185) suggested that the relationship between plasmid-positive cells and their plasmid-negative host is a consequence of the rate at which plasmid-free cells appear (Segregational instability) and the nature of any competitive advantage or disadvantage conferred by the plasmid on the host cells, and that the segregational instability of a plasmid is affected by several factors, which can be divided into three areas: host phenotype, plasmid phenotype and environmental conditions.

In 1986, Srienc et. al. (186) suggested that recombinant cells often exhibit genetic instability due to mutation or due to failure to transmit the recombinant DNA to progeny cells. Furthermore, in order to maximise the productivity of plasmid positive cells, the genetic and environmental parameters affecting stability and growth of

recombinant cells must be understood. One way to maximise the productivity of plasmid-positive cells in a bioreactor is to "design" the cells and the growth medium to give plasmid-positive cells a growth advantage using selective pressure (186).

Two main factors which are likely to affect the stability of a cloned gene in cells grown in chemostat culture were described by Walmsley et. al. (187), as follows:

1 - The spontaneous rate of segregation of the gene (or the plasmid on which it is borne);

2 - The impact of the plasmid on the host cell's maximum growth rate.

Competition between plasmid-negative and plasmid-positive cells in a population is dependent on whether one strain has a growth advantage over the other (185, 187). There are three competitive conditions which could occur in a chemostat:

1 - The presence of the cloned gene or plasmid is selectively neutral, (neither strain has a growth advantage);
2 - The plasmid-negative cells have a growth advantage over the plasmid-positive cells;
3 - The plasmid-positive cells have a growth advantage over the plasmid-negative cells.

In the absence of selection pressure, cells with

plasmids typically grow more slowly than cells without plasmids due to:

1 - A tendency for the copy number to increase per cell;

2 - An increase in gene expression from a plasmid gene;

The reduction in growth rate on plasmid-positive cells comparing with wild type may be attributed to the burden of extra plasmid DNA either due to plasmid copy number or plasmid size, and also the expression of high levels of foreign proteins.

2. THE AIMS OF THIS RESEARCH.

The aim of this project was the development, at an APPROPRIATE level of technology, of a microprocessor - controlled fermenter system and a full evaluation of it, in batch and chemostatic mode, as a tool for the investigation of the growth kinetics of a genetically engineered strain of Saccharomyces cerevisiae.

As was discussed in the introduction, developing countries like Brazil suffer both from a lack of resources to promote scientific research and development (in Universities and elsewhere) and from the commercial and political pressures of transnational corporation operations, which create and maintain situations of technological dependence in developing countries. Much imported machinery and equipment is set up in inappropriate environments which lack the necessary infrastructure for their continued effective use.

University environments in Brazil particularly lack an appropriate infrastruture to maintain high-tech instrumentation. Acquisition of such equipment, already make difficult by hard currency and bureauratic problems, creates severe problems of operation and maintenance caused by lack of technical backup, servicing and spare parts. In addition, much foreign computing equipment may not be imported a measure to protect the local microcomputer industry.

Self-sufficiency, then, becames extremely important. Reasonable scientific progress depends in the ability to do experiments, and this depends on being able to maintain, modify, develop and repair locally your scientific equipment.

This development of a microprocessor controlled fermenter involved, therefore, its design ab initio using Z-80 based electronic chips (an 8-bit computing system), chips which are, and will be for the foreseeable future, freely available at reasonable cost in Brazil.

The choice of biological system to evaluate the controlled fermenter also related to a biotechnologically important Brazilian industry, the sugar-alcohol industry. This old and traditional technology is an important part of the social fabric of parts of Brazil (particularly the North East) and, as an ingredient of the major Brazilian biotechnology initiative - The PROALCOOL program, is one ripe for a degree of process improvement aimed particularly at the microbiology of cane sugar fermentation to alcohol.

3. - MATERIALS AND METHODS

3.1 - Microorganisms

Three different strains of Saccharomyces cerevisiae were used in this project (table 3.1). Strain BC55, carrying the plasmid pCYG4, was developed by Nagasu and Hall (149) using the Neurospora crassa gene as a probe to screen a lambda library of yeast genomic DNA. They isolated a gene exhibiting strong homology to the Neurospora crassa gene and conferring NADP-GDH activity in yeast. This gene was subsequently cloned into the Escherichia coli-yeast shuttle vector CV13 (YEp13) BamHI site in Saccharomyces cerevisiae (gdh^- leu2 strain BC55) with subsequent selection for $LEU2^+$ transformants. This yeast shuttle vector (CV13) carrying the cloned fragments complements the gdh^- $leu2^-$ strain which directs substantial overproduction of NADP-GDH (gdh^+ $leu2^+$).

Table 3.1. Three different strains of Saccharomyces cerevisiae

Strain	Vector
Σ1278b	--------
BC55 (a gdh^- $leu2^-$ Bla^+)	pYE13
BC55 (a gdh^+ $leu2^-$ Bla^+)	pCYG4 (*)

(*) The strain BC55 transformed by the plasmid pCYG4 is the same strain BC55 with plasmid pYE13, but it has encoded the gene for NADP-GDH in the plasmid.

The strains were tested for NADP-GDH activity in vivo by comparing the growth rate of GDH^+ and gdh^- strains on minimum medium containing 2 mM d-histidine. The yeast strain BC55 transformed by plasmid pCYG4 was streaked on the minimum medium agar plate with and without 2mM D-histidine. After 3 days at 30°C, pCYG4 cells showed growth compared to gdh^- strain.

Saccharomyces cerevisiae wild type Σ1278b and Saccharomyces cerevisiae strains BC55 were obtained from A. Racher and Dr. J. R. Kinghorn (University of St. Andrews).

3.1.1. - Detection of plasmid in continuous culture.

Continuous culture were carried out using essentially the same minimal medium used by Nagasu and Hall (149), which does not have any extra supplements such as tryptophan and

leucine.

The presence of plasmid in the cells was checked using two different approaches.

1 - The determination of NADP-GDH activity in the cells. NADP-GDH activity may be a consequence of integration of plasmid into the genome, which was confirmed by Southern Blot analysis at a level of 10% with the plasmid YEp13 (193), otherwise, the activity present in the cells is correlated with NADP-GDH encoded in the plasmid.

2 - The level of penicillinase activity in the supernatant of the broth culture was measured. This method was based on the Chevallier and Aigle's method (194) which showed that plasmid carrying the Amp^R in a chimeric plasmid made of the 2μ DNA plasmid can produce penicillinase, and it can be detected by a halo around the penicillinase-producing strains (due to the reducing action of penicilloic acid), growing in a solid medium incorporating starch and iodine.

3.2 - Growth methods.

Both batch and continuous (chemostatic) growth modes were employed to study the behaviour of the strain Saccharomyces cerevisiae BC55 carrying the plasmid pCYG4.

3.2.1 - Batch Growth.

Batch experiments of Saccharomyces cerevisiae strain BC55 carrying the plasmid pCYG4 were carried out using two different media.

1 - The first medium used a supplement of leucine to eliminate the possibility of leucine becoming a limiting factor for the leucine auxotrophs (unselective medium).

2 - The second medium was designed without leucine (selective medium).

3.2.1.1 - Unselective Medium.

2.0 % Glucose
0.17 % Yeast Nitrogen Base
2 mM Ammonium sulphate
1mM L-Leucine

3.2.1.2 - Selective medium

2.0 % Glucose
0.17 % Yeast Nitrogen Base
2 mM Ammonium sulphate

3.2.1.3. Preparation of inocula.

The inoculum was prepared using the selective medium (described above) with 15mM ammonium sulphate to avoid ammonia being limiting. 50 ml of medium was inoculated in a shake flask, grown for 18h with shaking at 30°C and used to inoculate 550 ml medium in the fermenter vessel.

3.2.1.4. Preparation of the fermenter vessel.

The preparation of the fermenter vessel consisted of two steps:

3.2.1.4.1 - Calibration of the transducers.

This step consists of calibration of the transducers and other devices attached to the fermenter vessel (before the sterilisation procedures) following the description in section 4.5 (Interface Boards).

3.2.1.4.2 - Sterilisation Procedures.

The sterilisation of the fermenter vessel consists of a series of procedures as follows:

1 - The PCS will be turned on and tested for 10 min as follows: A - Temperature and oxygen will be adjusted to set-point values; B - Data from the fermenter will be collected each 10 seconds. After this test period the

system will be turned off.

2 - The transducer cables, silicon tubes (tubes for addition of nutrient, alkali and anti-foam into the fermenter) and the condenser tubes will be disconnected and closed.

3 - The central part of the heater will be pulled out from the fermenter vessel.

4 - The fermenter vessel will be autoclaved for 20 min at 121°C (15 psi) - WITH A PORT ON THE FERMENTER PLATE LEFT OPEN DURING THE STERILISATION.

5 - After sterilisation the open port will be closed immediately.

6 - The fermenter will be left in the autoclave for one hour or more to cool down.

3.2.1.5 - Specific Growth Rate.

The specific growth rate in batch culture was calculated using the following equation (195).

$$\ln X = \ln Xo + \mu t \qquad (1)$$

Where Xo is the biomass when t=0. The plot of lnX against time will be a straight line with a slope μ. The calculation was carried out using a computer-based linear

regression procedure. (POLYREG - See section 5.3.2).

3.2.1.6 - Mean generation time estimation.

The mean generation time (Gt or t_d = doubling time) was calculated using the method described in Pirt (195), following the equation below:

$$Gt = \frac{In2}{\mu} = \frac{.693}{\mu} \qquad (2)$$

3.2.2 - Continuous Culture.

3.2.2.1 - Introduction.

Those working with continuous culture techniques consider the pre-occupation of microbial geneticists and molecular biologists with batch cultures as both foolish and puzzling (196). Matin (197) has pointed out that continuous culture techniques (chemostatic) are vastly superior to batch culture for studies on microbial regulatory mechanisms.

The advantages of using continuous culture are that the cell environment can be rigidly controlled, a constant reproducible physiology can be maintained and it is essentially time independent, unlike in batch culture conditions (198). However, Tempest (199) pointed out a hysteresis effect on RNA content when cells were grown at progressively higher growth rates. He suggested that continuous culture was not always time independent.

A chemostat is basically a culture vessel having an input aperture for the influx of sterile nutrient medium from a reservoir and an overflow aperte for the efflux of medium, cells and cellular debris (200).

The system described in the section 3.2.1.4 was used in both continuous and batch experiments. Continuous experiments were carried out at dilution rates between 0.0125/h to up 0.15/h.

3.2.2.2 - Growth Limiting Substrate.

Three different growth limiting substrates were used in the chemostat experiments as follows:

3.2.2.2.1 - Carbon Limiting medium.

0.50 % Glucose

0.40 % Ammonium sulphate

0.17 % Yeast Nitrogen Base

3.2.2.2.2 - Nitrogen Limiting medium.

1.00 % Glucose

0.08 % Ammonium Sulphate

0.17 % Yeast Nitrogen Base.

3.2.2.2.3 - Glutamate Limiting medium.

1.00 % Glucose

0.21 % L-Glutamic acid

0.17 % Yeast Nitrogen Base.

3.2.2.3 - Preparation of the Inocula.

The carbon limiting medium was used as a medium in the inoculum preparation in all experiments in continuous culture. The procedures used were as described in section 3.2.1.3.

3.2.2.4 - Procedures.

Continuous culture experiments using the strain BC55 with plasmid pCYG4 were carried out in a chemostat system

using three different limiting substrates as follows.

1 - Carbon limiting (glucose as carbon source).

2 - Nitrogen limiting. Two different sets of experiments were carried out using different substrates.

A - Ammonium as nitrogen source.

B - Glutamate as nitrogen source.

All experiments were carried out under the same conditions: temperature 30°C, dissolved oxygen was maintained at 30% of air saturation, pH 5 controlled by addition of KOH (2M). These conditions were controlled by the Processor Control System (PCS), which also measured the concentration of biomass and the amount of KOH used to maintain the pH constant.

The continuous influx of substrate starts when the batch culture reaches two thirds of its exponential growth phase. During a chemostat experiment, which normally last about 45 generations, samples of 20ml from the washout were used to measure:

1. Enzyme activities: NADP-GDH, GOGAT, NAD-GDH, Penicillinase
2. Metabolite concentrations inside the cells: Ammonium, Glutamate.

3.2.2.5 - Maximum Specific Growth Rate.

The maximum specific growth rate was determined using the slope of a washout curve by the method of Pirt (121), using the following equation:

$$\ln X = (\mu m - D)t + \ln Xo \qquad (3)$$

Where the μm is the maximum specific growth rate. If we make dilution rate > critical dilution rate in the chemostat, the culture biomass decreases and washout occurs, following equation 3, whose slope (μm - D) gives the value of μm. Calculations were carried out using a linear regression program (POLYREG - see section 5.3.2).

3.3 - Measurement of Ethanol Concentrations

A gas-liquid chromatographic procedure was used to measure ethanol concentrations.

Measurements were carried out on a Pye Series 104 (Unicam Instruments, Cambridge, England) Gas Chromatograph. The column was packed with Porapak Q (Waters Associates, Inc., Milford, Mass., U.S.A.) 80-100 mesh. During the operation it was maintained at 200°C. Nitrogen was used as carrier gas at a flow-rate of 37 ml/min. The flame-ionization detector was operated at 250°C.

For calibration purposes, isopropanol was using as internal standard with a series of aqueous standards of ethanol in several concentrations from 1 to 9 mg/mg of isopropanol. For each standard, the ethanol/isopropanol peak-height ratio was calculated and plotted against the respective ethanol concentration. Figure 3.3 shows the relationship.

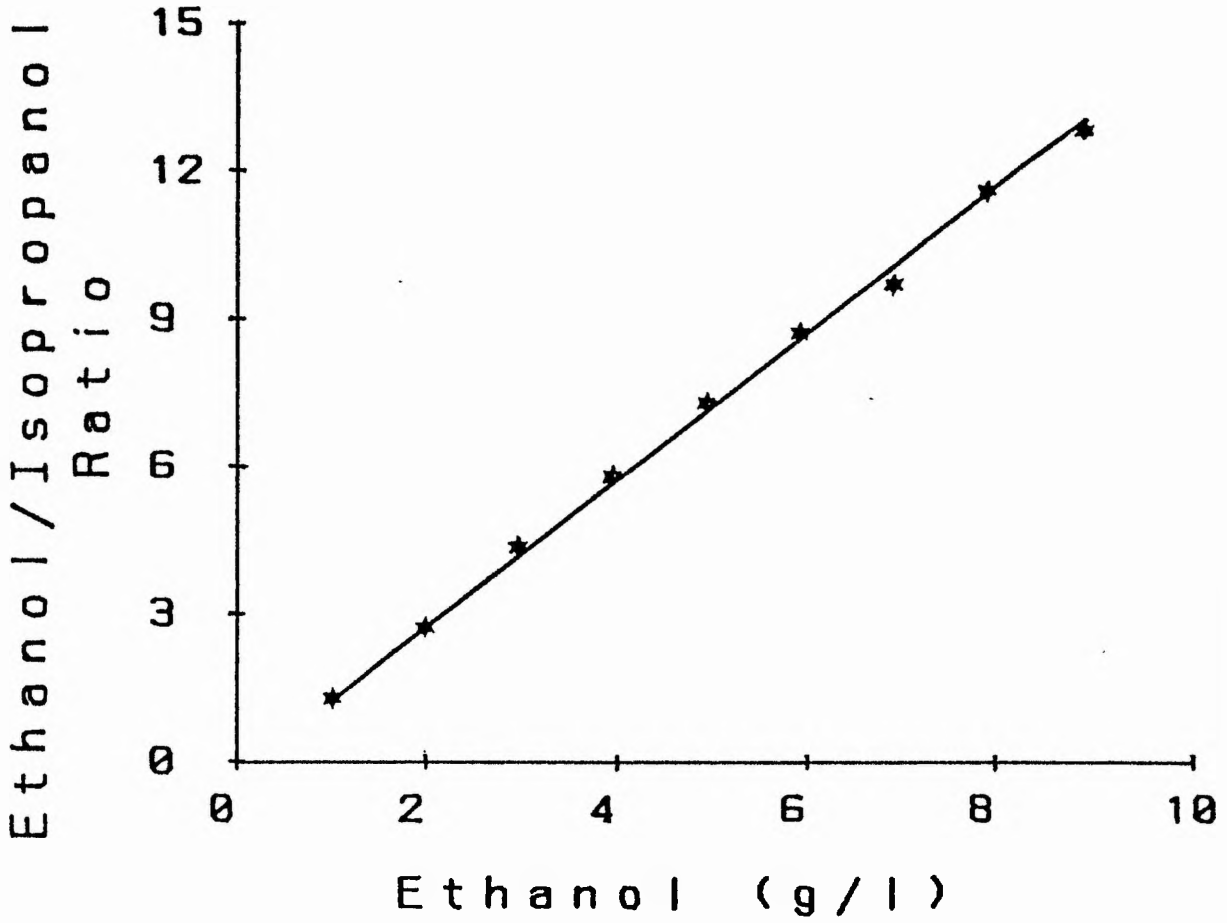


Figure 3.3. - Calibration plot of Ethanol

3.4 - Absorbance-Dry Weight Relationship

A culture of 600ml of Saccharomyces cerevisiae Σ1278b wild type grew aerobically for 24 hours in a selective medium. The whole culture was then centrifuged at 1000 rpm/10 min. The pellet containing cells was retained and washed twice with water before finally being resuspendend to a thick but pipettable solution. From this solution a series of suitable dilutions were made and their absorbances recorded at 650nm with a Pye SP 600 series 2 (Unicam Instruments, Cambridge, England) spectrophotometer against a distilled water blank. Three 1 ml samples of this solution were dried for 18 hours (oven 120 °C) and then weight accurately until the readings were constant. Since the accurate weight of 1ml of cells was known the equivalent dry weight of the dilutions could be calculated. A calibration plot was constructed, plotting the equivalent dry weight against absorbance readings. Figure 3.4 presents these results.

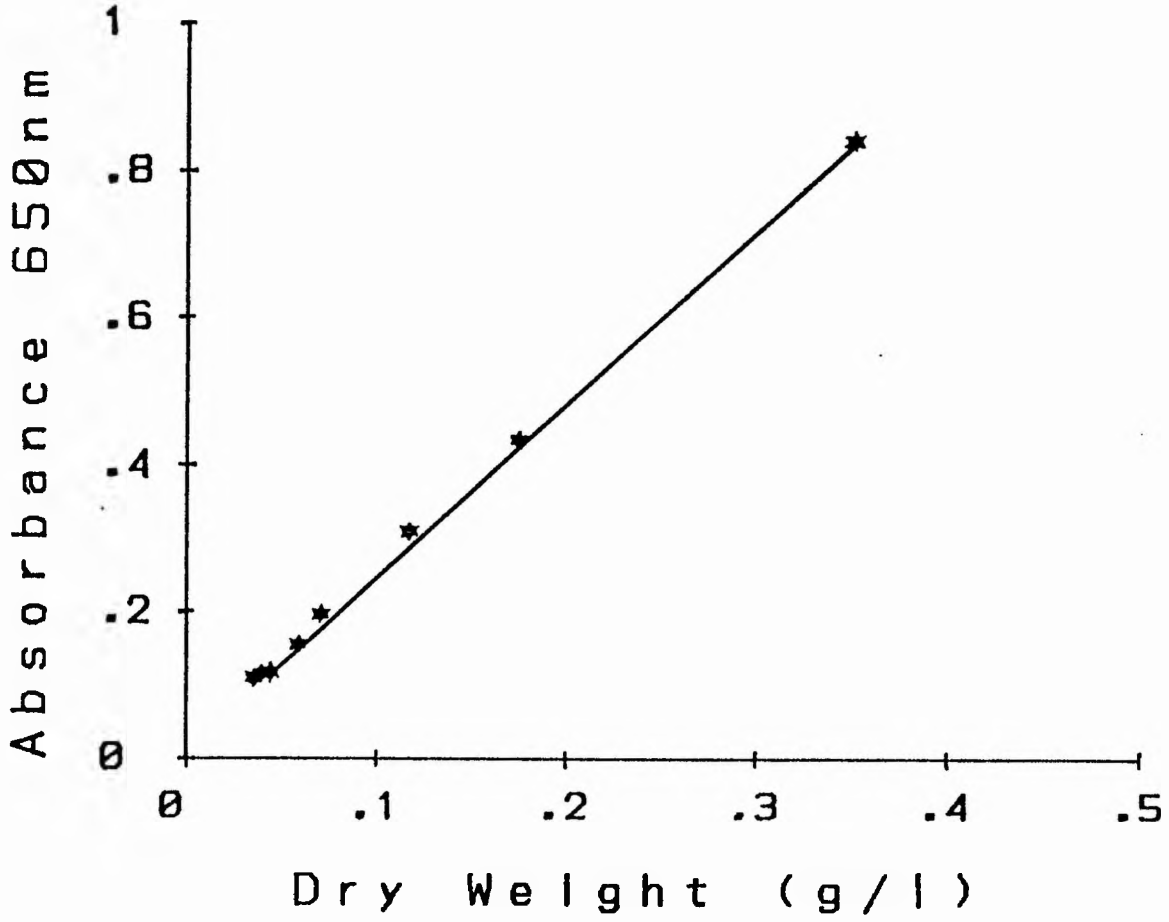


Figure 3.4 Absorbance-Dry Weight Relationship plot.

3.5 - Measurement of Glucose

3.5.1 - Reducing Method using 3,5-dinitrosalycilic acid (DNSA)

The procedure to determine reducing sugars were carried out as described in Bernfield (201).

3.5.2 Enzymatic Determination of Glucose (SIGMA)

The enzymatic determination of glucose was carried out using a kit based on the glucose oxidase-peroxidase system which measures microgram quantities of glucose in biological fluids (Sigma Chemical LTD (No.510, England).

3.6 - Protein Determination

3.6.1 Biuret method

Protein measurements using the Biuret method were carried out as described by Llyne (202).

3.6.2 Bradford method.

This method is a rapid and sensitive one for the measurement of microgram quantities of protein utilising the principle of protein-dye binding. The method which involves the binding of Coomassie Brilliant blue G-250 to protein is described below following Bradford (203).

3.6.2.1 Reagent.

Dissolve 100mg Coomassie Brillant blue G in 500ml 95% ethanol. To this solution 100ml 85% (w/v) phosphoric acid is added. The resulting solution is diluted to a final volume of 1 litre.

3.6.2.2 Assay.

1 - 0.1ml of protein sample is added to 5ml of Bradford's regent.
2 - Wait 10 minutes at 20°C - 25°C (room temperature).
3 - Absorbances are measured at 595nm against a blank buffer.

Figure 3.6.2 shows the calibration plot.

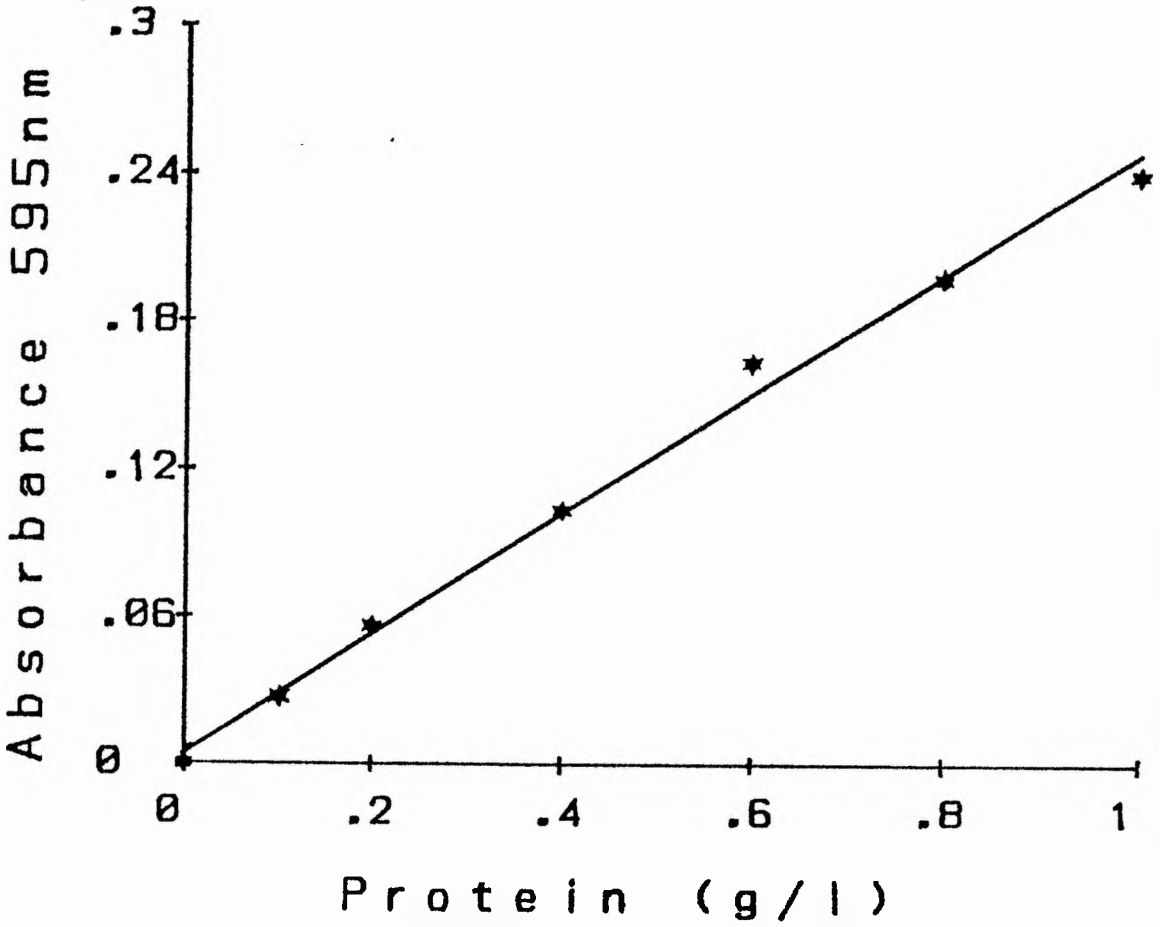


Figure 3.6.2. - Calibration plot of Protein using the Bradford method.

3.7 - Determination of Extracellular Penicillinase Activity.

Penicillinase activity may be determined in a simple and rapid fixed-time assay in which the product of penicillin hydrolysis (penicilloic acid) reduces iodine to iodide - a reaction which is detected colorimetrically (204, 205, 206, 207, 208).

3.7.1. Reagents:

1 - Stock iodine solution.

0.32N iodine and 1.2M potassium iodide, prepared by dissolving 20.3g of resublimed iodine and 100g of potassium iodide in 500ml of distilled water.

2 - Iodine reagent.

5 ml of stock iodine is added to 95 ml of acetate buffer pH 4.0 (Acetate buffer - 980g of anhydrous sodium acetate adjusted to pH 4.0 with acetic acid in 2 litres of distilled water).

3 - 20,000 units of Penicillin-G solution (Benzylpenicillin - potassium salt) (Sigma).

3.7.2. Preparation of the sample

5ml of culture broth supernatant was used as a penicillinase sample. It was obtained by centrifugation (13,000 rpm/2min - Eppendorf microcentrifuge). The pH was adjusted to 7.0 by using 0.1M phosphate buffer.

3.7.3 Assay:

1 - 2.5ml of penicillinase-containing sample (pH 7.0) was equilibrated at 30°C, then 0.5ml penicillin G solution pH 7.0 (0.1M phosphate buffer) (20,000 U/ml) was added.

2 - After 30min the reaction was stopped by the addition of iodine reagent (5ml), followed immediately by rapid mixing.

3 - In addition, three test tubes (a to c) were prepared simultaneously as controls.

a - Sterile medium pH 7.0 (phosphate buffer, 0.1M) (2.5ml) plus penicillin G solution (0.5ml);

b - 2.5ml of the sample solution plus sterile medium pH 7.0 (0.5ml);

c - Sterile medium pH 7.0 (3.0ml).

4 - Absorbances (A) were taken at 490nm of the three controls and the sample.

Aa - Absorbance of the control a

Ab - Absorbance of the control b

Ac - Absorbance of the control c

As - Absorbance of the sample

The final absorbance (FA) was calculated following the equation below.

$$FA = Aa-(Ac-Ab)-As$$

5 - The enzyme activity was determined using a calibration plot (figure 3.7) carried out measuring the activity of a penicillinase (β-lactamase I; EC 3.5.2.6, Bacillus cereus type I. - Sigma) in different concentrations following the procedures described above.

The calibration plot is shown in figure 3.7.
A unit of penicillinase activity is that quantity of enzyme which hydrolyzes 1μmole of penicillin G (benzyl penicillin) per hour, at 30°C, pH 7.0.

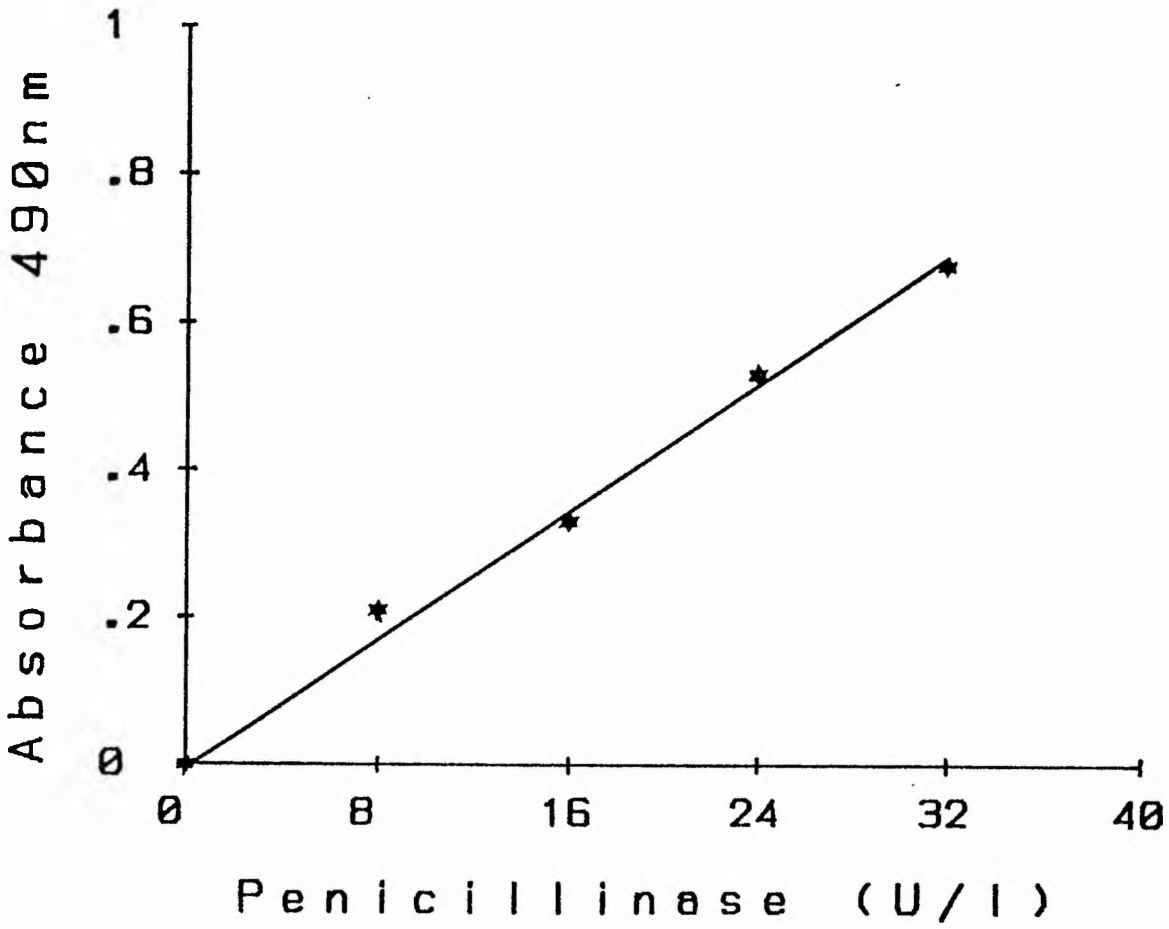


Figure 3.7 - Calibration plot of Extracellular Penicillinase Activity.

3.8 - Determination of Concentrations of Intracellular Metabolite Pools (Ammonia and L-Glutamate)

3.8.1. - Extraction of Intracellular metabolites

For the estimation of metabolic pools, 3ml cell culture aliquots were rapidly centrifuged (Eppendorf - microcentrifuge 13,000 rpm/3 min) , then washed three times with distilled water, then immediately suspended in 1.5 ml of Tris HCl buffer (0.5M; pH 8.0) using a Eppendorf tube and placed in a autoclave (121°C, 10 psi per 10 min). The samples were then cooled in an ice bath, and cell debris was removed by centrifugation for 3 min (Eppendorf - microcentrifuge 13,000 rpm/3 min). The clear supernatant was decanted and used for analysis.

The enzymatic determination of ammonia and L-glutamate using (NADH) and glutamate dehydrogenase (NADP-GDH) were carried out as described by Bergmeyer (209).

3.9 - Enzyme Assays

3.9.1 - Reagents

1 - Extraction Buffer (EB)0.15% triton X-100 in Tris-HCl, pH 8.0 (100mM).

2 - 2-oxoglutarate (0.2M in EB).

3 - NADPH (1mg/ml in EB).

4 - NADH (1mg/ml in EB)

5 - Ammonium chloride (0.2M in EB).

6 - L-Glutamine (0.04M in EB).

All the reagents were produced by Sigma Chemical Company.

3.9.2 - Preparation of extracts

A 3ml aliquot of cell culture was centrifuged (Eppendorf: 3 min / 13,000 rpm). The pellet was resuspended in 1.5ml extraction buffer, and glass beads (0.45 - 0.50 mm) were added up to approximately the same volume as the pellet. Cells were broken with a vortex mixer for 3 min. Supernatants were removed after centrifugation (Eppendorf - microcentrifuge 13,000 rpm/min) and used as a sample for the enzyme assay reactions.

3.9.3 Calculation of enzyme activities.

The activity of the three enzymes were carried out using the formule described below .

$$\text{Activity} = \frac{(\text{Vol. of assay mix (ml)}) \times (\text{Ex./mim})}{6.22 \times \text{Vol. sample (ml)}} = \text{U/ml}$$

Where Ex is the rate of NADPH decrease per minute at 30°C.

$$\text{Specific activity} = \frac{\text{Activity (U/ml)}}{\text{Protein concentration (mg/ml)}} = \text{U/mg}$$

3.9.4 - NADP - dependent glutamate dehydrogenase.

The NADP-dependent glutamate dehydrogenase activity was determined measuring the decrease of NADPH at 340 nm using a molar extinction coefficient of 6.22 $cm^2/\mu mole$, following the reaction below:

$$NH_4^+ + NADPH + 2\text{-oxoglutarate} \xrightarrow{\text{NADP-GDH}} \text{L-Glutamate} + NADP$$

3.9.4.1 - Assay

Pipette into cuvettes:	
Sample	0.20 ml
Extraction buffer	2.60 ml
Ammonium Chloride	0.10 ml
2-oxoglutarate solution	0.10 ml
(wait appx. 3-4min)	
NADPH solution	0.20 ml

One Unit of activity was defined as the amount of enzyme which produced 1µmol of NADP in 1 min at 30°C in the standard assay conditions (30°C, pH 8.0).

3.9.5 - NAD - dependent glutamate dehydrogenase.

The NAD-dependent Glutamate dehydrogenase activity was determined measuring the decrease of NADH at 340 nm, using a molar extinction coefficient of 6.22 $cm^2/\mu mole$, following the reaction below:

$$\text{L-Glutamate} + \text{NAD} \xrightarrow{\text{NAD-GDH}} NH_4^+ + \text{NADH} + \text{2-oxoglutarate}$$

3.9.5.1 - Assay

Pipette into cuvettes:	
Sample	0.20 ml
Extraction buffer	2.60 ml
Ammonium chloride	0.10 ml
2-oxoglutarate solution	0.10 ml
(wait appx. 3-4min)	
NADH solution	0.20 ml

One Unit of activity was defined as the amount of enzyme which produced 1µmol of NAD in 1min at 30°C in the standard assay method.

3.9.6. GOGAT (Glutamine (amide): 2-oxoglutarate amino-transferase oxido-reductase (NADP))

GOGAT activity was determined measuring the rate of NADPH decreases at 340 nm, using a molar extinction coefficient of 6.22 $cm^2/\mu mole$, following the reaction below:

$$\text{L-Glutamine} + \text{NADPH} + \text{2-oxoglutarate} \xrightarrow{\text{GOGAT}} \text{2L-Glutamate} + \text{NADP}$$

3.9.6.1 - Assay

Pipette into cuvettes:	
Sample	0.20 ml
Extraction buffer	2.60 ml
L-Glutamine	0.10 ml
2-oxoglutarate solution	0.10 ml
wait appx. 3-4min	
NADPH solution	0.20 ml

One Unit of activity was defined as the amount of enzyme which produced 1µmol of NADP in 1 min at 30°C in the standard assay conditions (30°C, pH 8.0).

4. - The Processor Control System (PCS)

4.1 - Introduction.

The Processor Control System consists of three different sub-systems (boards):

1 - Main boards consisting of 3 distinct boards mounted horizontally in the PCS frame.

2 - Auxiliary boards, which are a group of four boards fitted vertically on the left hand side of the PCS frame.

3 - A connection block fitted on the top of the PCS frame used to connect the PCS with the video terminal, with sensors from the fermenter system and with other microcomputer systems (See figure 4.1a). Figure 4.1 shows the diagram of of the front part of the PCS with its three main blocks.

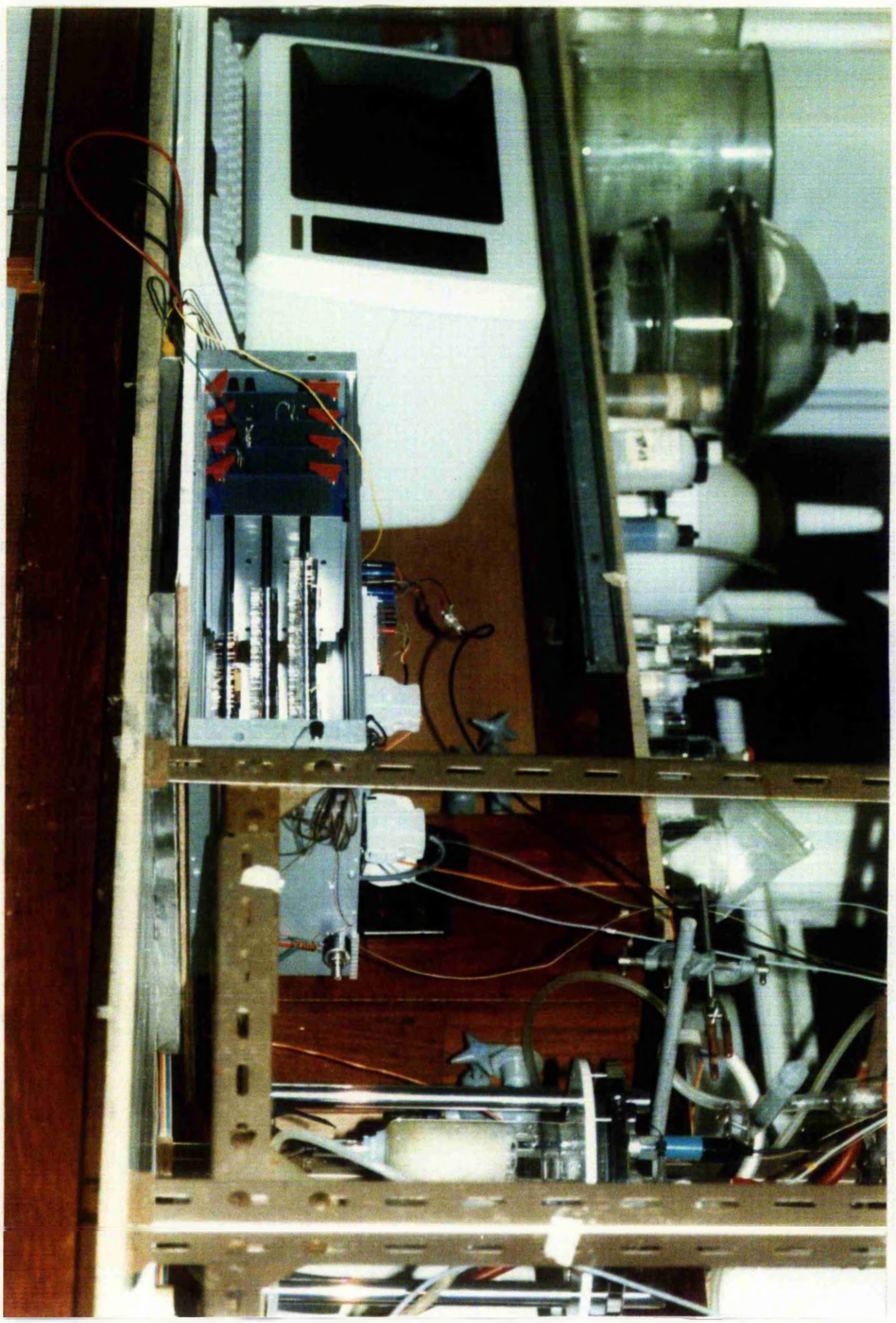
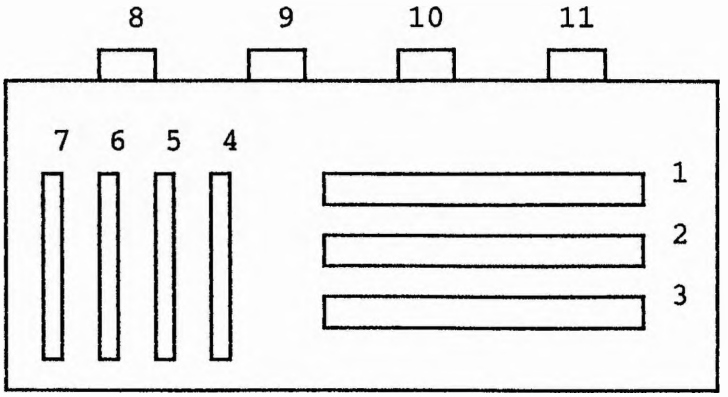
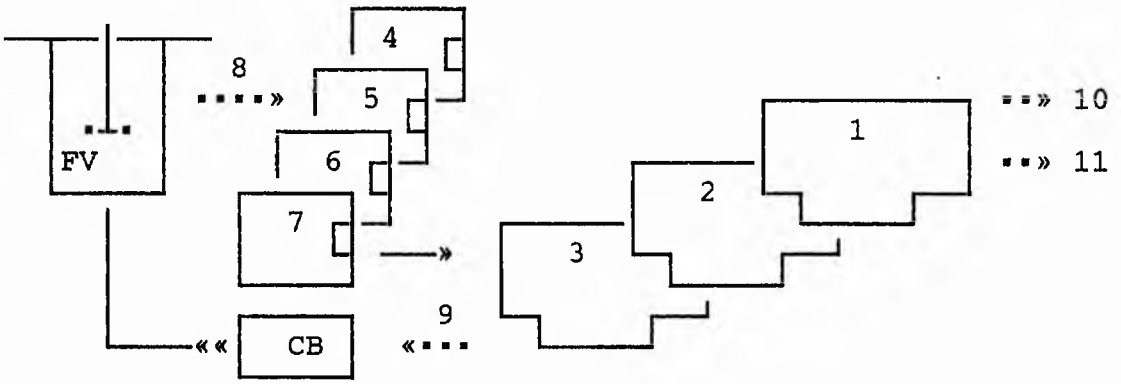


Figure 4.1a - Photograph of the front view of the PCS.

Figure 4.1 - Diagram of the front view of the PCS





FV - Fermenter vessel

CB - Control box

Board Configurations:

First Block (Horizontol boards)

1-Central Processor Board (CPU-Board)

2-Memory Board

3-Analog/Digital Converter (A/D) and (ON/OFF) Switch board (ADS-board).

Second Block (Vertical Boards)

4-pH Interface board

5-Oxygen Interface board.

6-Temperature Interface board

7-Biomass Interface board

Third Block (On the top and ▪▪▪» or «▪▪▪)

8-Connection between Sensors in the Fermenter Vessel (FV) and the Interfaces.

9-Connection between A/D and the Control Box.

10-Connection between the PCS and the Video terminal (RS323C link).

11-Connection between the PCS and another microcomputer

A printed circuit high density, square S100 card (RS 434-633) was used to assemble the integrated circuits in the three main boards (first block), and the square pad (RS 434-605) was used in the construction of the boards in the second block. The connecters 10 and 11 are 'D' connecters type 25-way (RS 467-891), the connecter 9 is a DIN 41612 - class III - 96 way (RS 471-480). The connecter 8 is a locally constructed box with eight sockets (RS 444-450) used to connect with 'banana' plugs from the sensors.

The PCS was constructed using the wire-wrapping technique which is described below.

4.2 - Wire-Wrapping Technique

The wire-wrapping technique is an inexpensive and flexible technique for the construction of small electronic circuits such as microcomputer systems.

4.2.1 - Components and Tools:

1 - Wire-wrapping tool which is a combined wire stripping, wrapping and unwrapping hand tool designed to strip, wrap and unwrap wire on a 0.85mm terminal pin (RS).

2 - Wire-wrapping sockets (RS).

3 - Wire-wrapping high density S-100 board (RS).

4 - Wire-wrapping terminal pins (F).

5 - Wire for wire-wrapping (RS).

6 - Soldering tool (RS).

RS - RS Components, Corby, England.

Farnell - Farnell Electronic Components, Leeds, England

4.2.2 - Procedure

4.2.2.1 - Stripping and Wrapping

1 - Strip the wire, using the wire-wrapping tool, of about one inch of insulation.

2 - Insert the stripped wire into the end of the wire-wrap insert the tool over the pin of the socket or of the terminal pin.

3 - Press it down gently and rotate manually to wrap the pin.

4 - Normally each socket pin or a terminal pin should allowed three levels of wrapping without contact problems.

5 - The single terminal pins should be soldering on the wrapping side of the board.

4.2.2.2 - Unwrapping

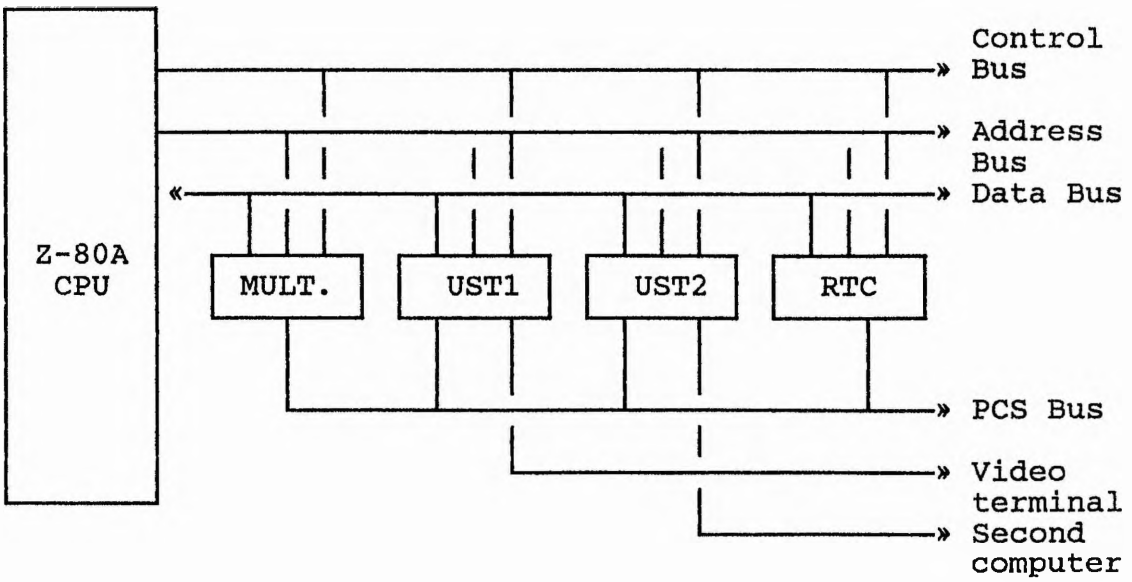
Insert the other end of the wrap-wrapping tool over the socket pin or the terminal pin and rotate anti-clockwise until the wire is removed, with help from a pair of long-nosed tweezers.

4.3 - The Central Processor Unit Board (CPU-Board)

4.3.1 - Introduction.

The CPU-board, heart of the PCS, consists of several integrated circuits (I.Cs.)(see figuree 4.3a and 4.3c) and two quartz crystals, two DIL end stackable switches and a few resistors and capacitors. Its function is to process the digital information from the A/D board or from the remote serial port (connecters 10 and 11) RS232C link, then to store the results in the memory board or send them through the connecters 10 or 11 (RS232C) to another computer for further data analysis. The PCS also controls parameter conditions from the fermenter comparing data from the transducers with set point data stored in the memory. Figure 4.3b shows the block diagram of this board.

Figure 4.3b - Block Diagram of the CPU-board in the CPS.



Components description:

CPU - Central Processing Unit (210,211,212)

MULT. - Multiplexer SN74LS154N (213)

UST1 - Universal Synchronous/Asynchronous Receiver/Transmitter 1 (USART 1) (214)

UST2 - Universal Synchronous/Asynchronous Receiver/Transmitter 2 (USART 2) (214)

RTC - Real Time Clock

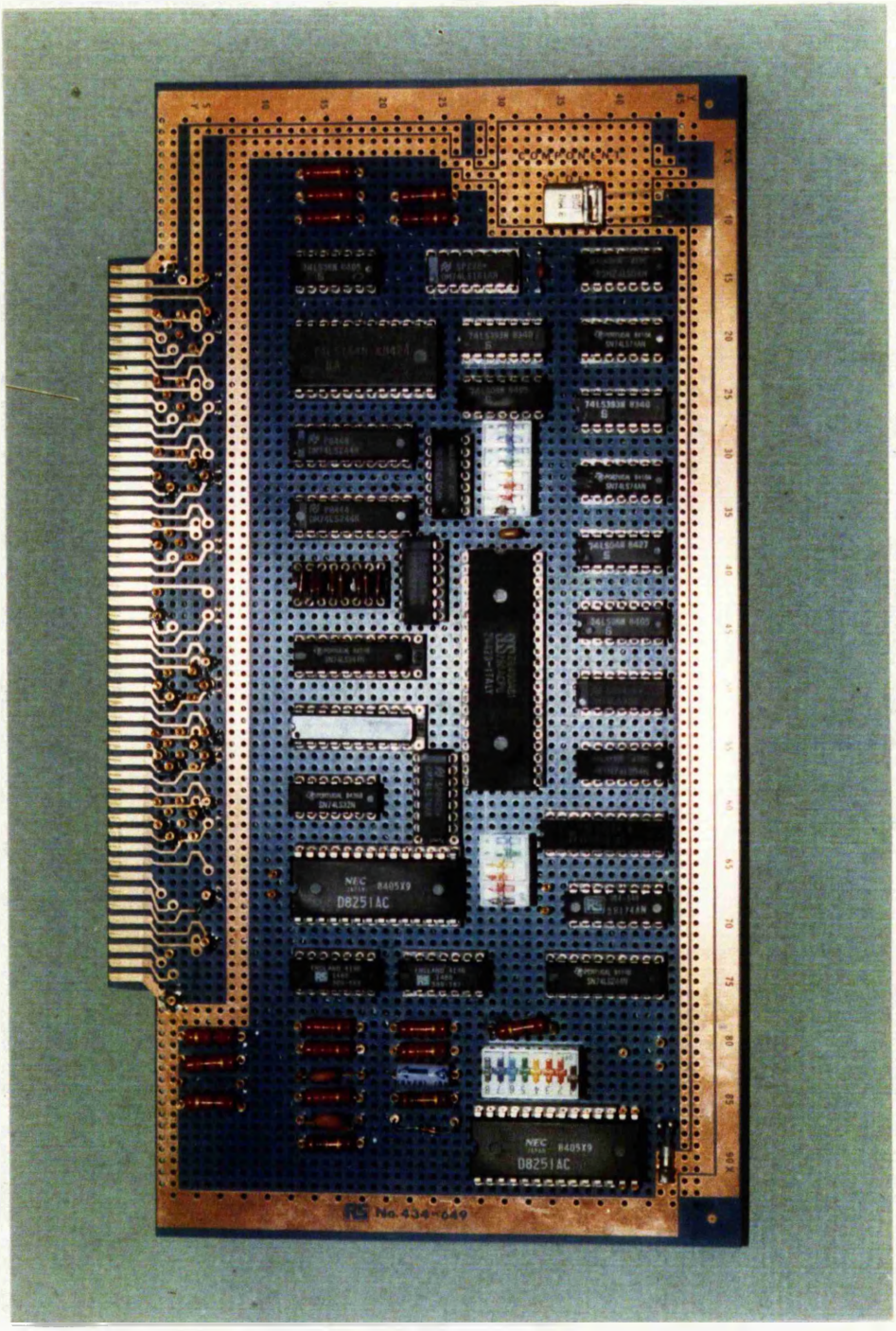
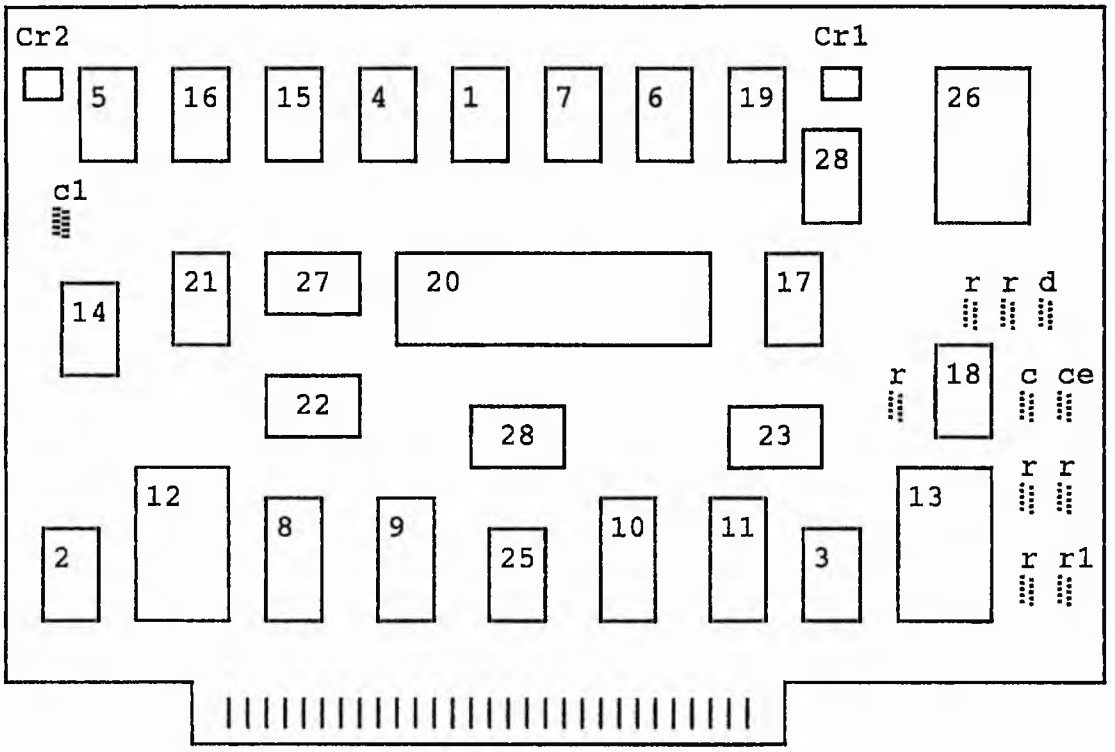


Figure 4.3c Photograph of the CPU Board of the PCS.

Figure 4.3a - Diagram of CPU-Board of the PCS



Component Description:

1 - SN74LS08N - Quad 2-Input AND Gate (T)

2 - SN74LS08N - Quad 2-Input AND Gate (T)

3 - SN74LS32N - Quad 2-Input OR Gate (T)

4 - SN74LS04N - Hex Inverter (T)

5 - SN74LS04N - Hex Inverter (T)

6 - SN74LS04N - Hex Inverter (T)

7 - SN74LS32N - Quad 2-Input OR Gate (T)

8 - SN74LS244N - Octal bus/Line Driver (N)

9 - SN74LS244N - Octal bus/Line Driver (N)

10 - SN74LS244N - Octal bus/Line Driver (N)

11 - SN74LS244N - Octal bus/Line Driver (N)

12 - SN74LS154N - 4-Line to 16-Line Decoder/Demultiplexer (T)

13 - P8251A - Programmable Communication Interface (N)

14 - DM74LS161AN - Synchronous 4-bit Binary Counter (N)

15 - SN74LS393N - Dual 4-bit Binary Ripple Counter (T)

16 - SN74LS74AN - Dual D positive Edge Triggered Flip/Flop with Clear (T).

17 - DS1488 - Quad Line Driver (N)

18 - DS1489 - Quad Line Receiver (N)

19 - RS 58174 - Real Time Clock I.C. (N)

20 - CPU Z-80 - Central Processor Unit (Z-80A) (Z)

21 - SN74LS08N - Quad 2-Input AND Gate (N)

22 - SN74LS04N - Hex Inverter (T)

23 - DM74LS74AN - Dual D-Type Edge-triggered Flip-Flop.(N)

24 - DM74LS74AN - Dual D-Type Edge-triggered Flip-Flop.(N)

25 - Resistors 1KΩ (RS)

26 - D8251AC - Programmable Communication Interface (I)

27 - DIL End Stackable Switches (ON/OFF) (F)

28 - DIL End Stackable Switches (ON/OFF) (F)

Cr1 - Quartz Crystal 32,768 KHz (F)

Cr2 - Quartz Crystal 8.0 MHz (F)

c - Ceramic disc capacitors 1000 pF, 63 volt (F)

ce - Electrolytic capacitor CEA series 10 µF, 25 volts (F)

c1 - Ceramic Plate capacitor 33 pF, (F)

r - High stability carbon film resistors 1KΩ

r1 - High stability carbon film resistor 100Ω.

The CPU-board was assembled on a S100 high density wire-wrapping board (RS). The Wire-wrapping pins (17.85 mm length) were used to connect the resistors and capacitors

with the wire wrapping terminals on the board (F).

(F) Farnell Electronics Components Limited (Leeds, England).

(T) Texas Instruments (Texas, U.S.A.).

(N) National Semiconductor Corporation (NSC) (California, U.S.A.).

(RS) RS Components (Corby, England).

4.3.2 - Principal Circuits in the CPU-Board.

4.3.2.1 - The Z-80A microprocessor.

The Z-80A microprocessor (210, 212, 213) is a third generation single-chip microprocessor with exceptional computational power. It is a 40-pin (40 legs, 20 per side) semiconductor integrated circuit (I.C) about 2 inches long by half an inch wide and it has the following main features:

A - The Z-80A has 158 instructions. The 78 instructions of the 8080A are included as a subset permitting total software compatibility.

B - It is TTL compatible.

C - There are three modes of high speed interrupt processing: 8080 compatible, non-Z-80A peripheral device, and Z80 family peripheral with or without 'daisy chain'.

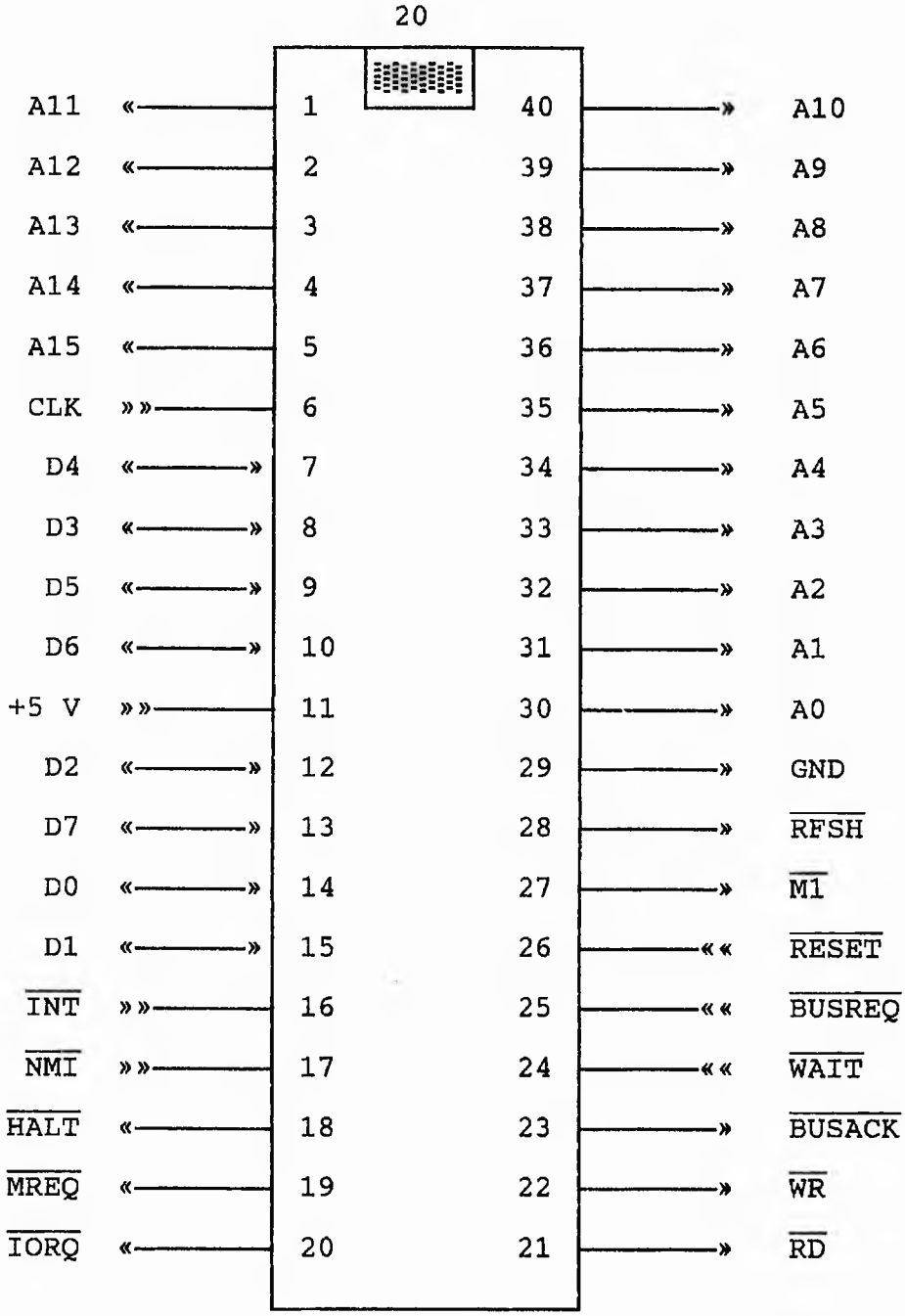
D - It has double sets of both general-purposes and flag registers of the 8080A.

E - It has two 16-bit index registers which facilitate program processing of tables and arrays.

F - It has an automatic dynamic RAM refresh circuit.

The pin configuration of the Z-80A microprocessor is described in figure 4.3.2.1.

Figure 4.3.2.1 - Z-80A Pin configuration



Pin Names:

A0-A15 Address Bus (unidirectional (output only), active at logic 1) formatting a 16 bit address. It provide addresses for memory data and also for input/output (I/O) devices (i.e. Analog/Digital converter).

D0-D7 Data Bus (active at logic 1) constitutes an 8-bit bidirectional data bus, used for data exchanges with memory and I/O devices.

$\overline{\text{BUSACK}}$ Bus Acknowledge , indicates to the requesting device that the CPU address bus, data bus, and control signals such as $\overline{\text{RD}}$, $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$ and $\overline{\text{WR}}$ have entered their high impedance states. It is very useful, because the external circuit can now control these lines.

$\overline{\text{INT}}$ Interrupt Request (active at logic 0) - an I/O device. The CPU honours a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled.

$\overline{\text{NMI}}$ Non-Maskable Interrupt (active at logic 0). It has a higher priority than $\overline{\text{INT}}$. It is recognised at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at memory location 0066H.

$\overline{\text{BUSREQ}}$ Bus Request (active at logic 0). It has a higher priority than NMI and is always recognised at the end of the current machine cycle. It is a very important control, because it forces the CPU address bus, data bus, and control signals $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ to go to a high impedance

state so that other devices can control these lines.

$\overline{WR}$ Write (active at logic 0), normally it is used to write to the memory. $\overline{WR}$ indicates that the CPU data bits hold valid data to be stored at the addressed memory or I/O device.

$\overline{RD}$ Read (active at logic 0), used to read memory data and also to read data from I/O devices.

$\overline{IORQ}$ I/O Request (Active at logic 0) indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation, is generated at the same time as $\overline{M1}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

$\overline{M1}$ Machine Cycle (active at logic 0), concurrently with $\overline{MREQ}$ it indicates that the current machine cycle is the opcode fetch cycle of an instruction execution, and also M1 with IORQ indicates an interrupt acknowledge cycle.

$\overline{WAIT}$ Wait (active at logic 0) indicates that the memory or an I/O is not ready for a data transfer. The CPU continues to produce a wait signal as long as this signal is active.

$\overline{RFSH}$ Refresh (Active at logic 0), together with $\overline{MREQ}$, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

$\overline{RESET}$ Reset (active at logic 0) is an input signal to the CPU that indicates that power to the CPU has been just turned off, or that the CPU should be reset. The reset causes the CPU to clear the Program Counter (PC) register, and after a short pause, start execution from memory location 0000H. It

is extremely important for initiation of the system because it allows the CPU to start from a known point. When the power is first turned on, the voltage at pin 26 (RESET) is near 0 V and it gradually builds up as the capacitor becomes fully charged.

$\overline{\text{HALT}}$ Halt state (Active at logic 0), indicates that the CPU has executed a $\overline{\text{HALT}}$ instruction and is waiting for either a non-maskable or a maskable interrupt (which the mask enabled) before operation can resume.

$\overline{\text{MREQ}}$ Memory Request (active at logic 0) stands for memory request and is at zero only when a memory read or write operation is being performed.

4.3.2.1.1 - CPU Registers

The most essential part of the Z-80A and any other microprocessor, as far as data handling is concerned, are the CPU registers. These are simple storage cell or memory locations in the CPU. Like other memory locations in many microcomputer systems, they are organized into: A - General prupose registers; B - Special purpose registers; C - Interrupt status flip-flop; D - Arithmetic and Logic Unit (ALU); E - Instruction register. They are further described below.

A - First Group - General Purpose Registers.

There are paired sets of 8-bit registers: The registers are the accumulator (A,A'), the Flag (F,F'), and six general-purpose registers (B,B', C,C', D,D',

E,E', H,H' and L,L'), as well as this set, the six individual 8-bit registers will be contained together in three 16-bit registers (AB,CD and HL registers). At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set.

B - Second Group - Special Purpose Registers.

This group consist of six registers with specific functions.

B1 - Program Counter (PC) - is related to the external memory. It is a 16-bit register that points to the current instruction that is being executed. The PC is automatically incremented after its contents have been transferred to the address lines. When a program jump occurs the new value is automatically placed in the PC, overriding the incrementer.

B2 - Stack Pointer (SP) as with the PC this register is also related with the external memory and holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in, first-out (LIFO) file.

B3 - Two Index Registers (IX and IY). These index registers were not in the 8080 instruction set and have been added to the Z-80A set to enable a type of addressing capability called indexing. This allows the IX and IY registers to be used as a type of indirect pointer similar to the HL or

other register pair.

B4 - Memory Refresh Register (R) is an 8-bit register used to provide a refresh address for a type of microcomputer memory called Dynamic memory. Because the PCS memory does not need to be continuously refreshed, this register has not been used.

B5 - Interrupt Page Address Register (I) is an 8-bit register used to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. It allows the CPU to achieve any memory location in response to an interrupt with absolute minimal access time to the interrupt routines.

C - Third Group - Interrupt Status Flip-Flop

This group consists of two interrupt status flip-flops which assist in identifying the interrupt mode at any particular time. These are the interrupt Enable (IFF_1-IFF_2) and the Interrupt Mode (IMFa-IMFb)

D - Arithmetic and Logic Unit (ALU)

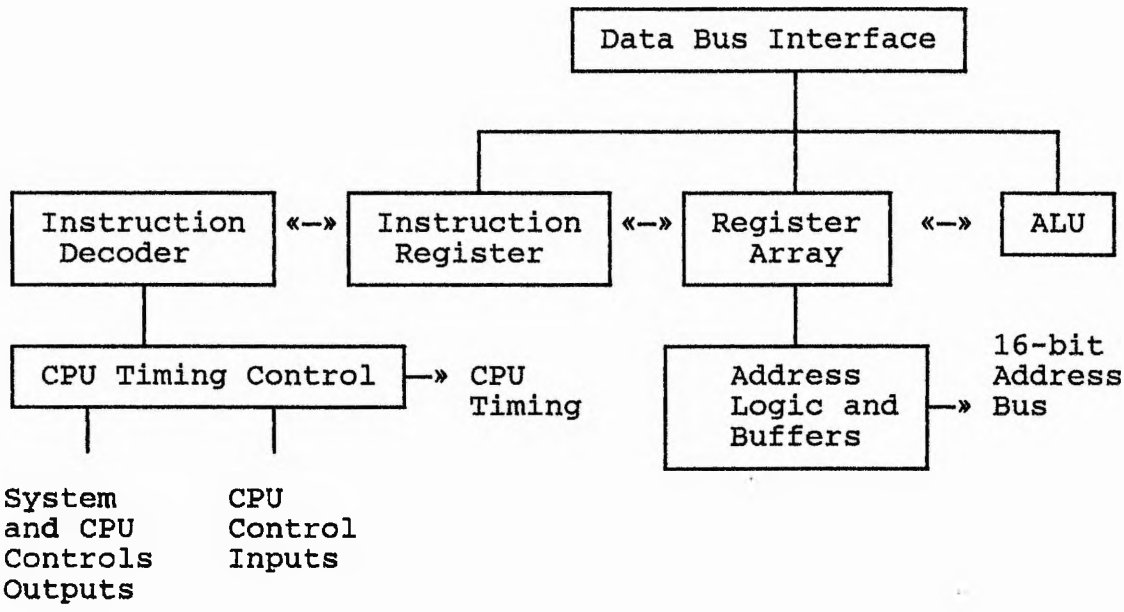
One of the most important functions of the Z-80A CPU is to perform arithmetical operations, that is to add or subtract or to carry out logical operations such as logical AND, logical OR, logical exclusive OR or the comparison of bits.

E - Instruction Register.

This register is used to supply all the control signals necessary to read or write data from or to the register, control the ALU and provide all required external control signals.

Figure 4.3.2.1.1 represents the block diagram of the Z-80A CPU, where the exchange of communication between the external devices and the internal operations of the microprocessor is shown.

Figure 4.3.2.1.1 - Block diagram of the Z-80A CPU



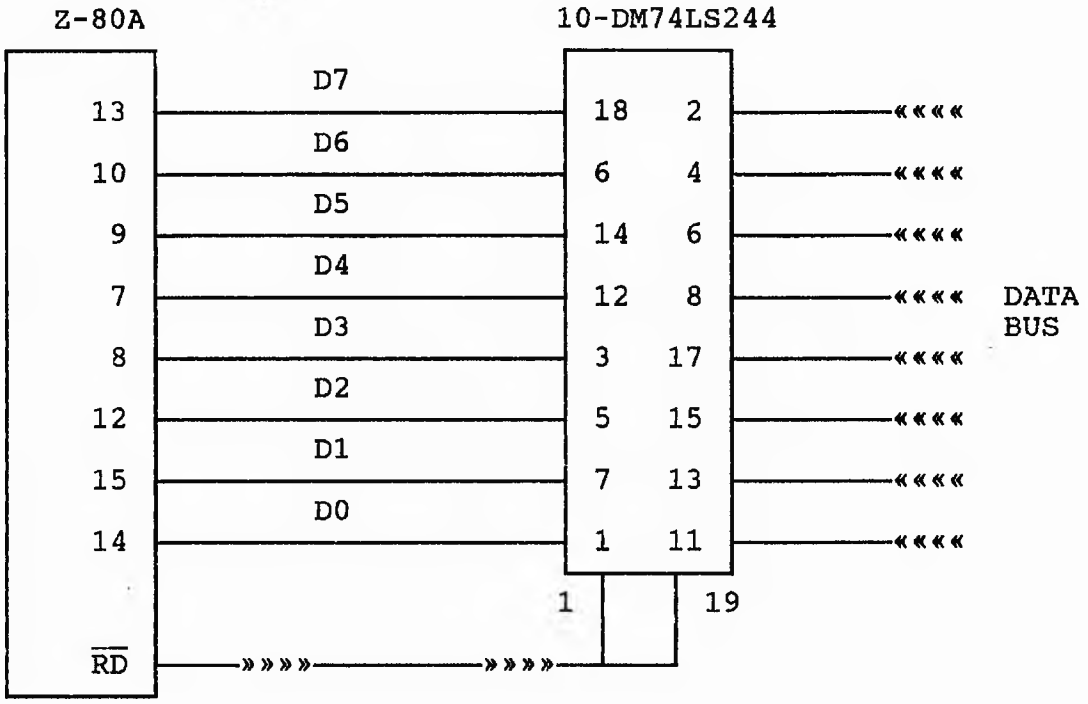
4.3.2.2 - CPU connections on the CPU Board

4.3.2.2.1 - I/O Data

The I/O data is transmitted by two octal bus/line drivers (DM74LS244, no. 10 and 11 - see figure 4.3a). They are monolithic complementary MOS (CMOS) integrated circuits with tri-state outputs. These outputs have been specially designed to drive highly capacitive loads such as bus-oriented systems. Each has 8 I/O lines, where every four lines are activated by one non-inverting control line and this is activated by a read signal from the CPU (see figures 4.3.2.2.1A and 4.3.2.2.1B, input and output, respectivaly).

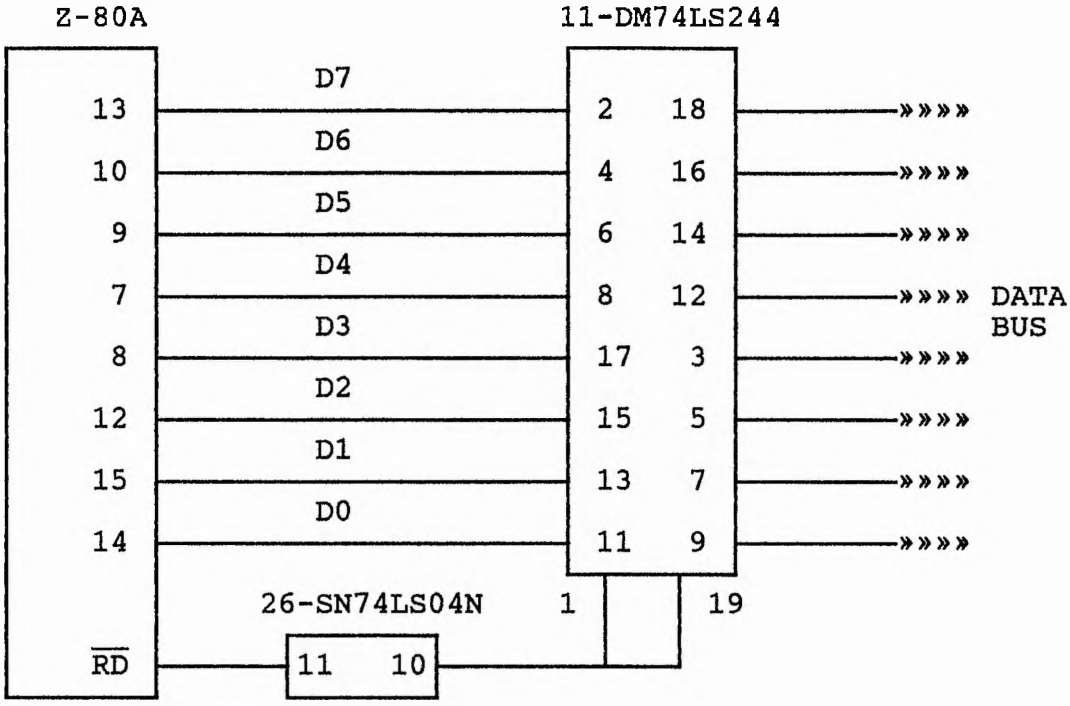
Input Data - Figure 4.3.2.2.1a shows the input lines from the system to the Z-80A. The I.C. DM74LS244 (no. 10, in figure 2.4a) sends data from the data bus to the CPU. Its function is controlled by two inputs pins (1 and 19) which are under the control of the RD line from the CPU, so when the RD line is activated (logic 0) the DM74LS244 is also activated and the data go to the CPU.

Figure 4.3.2.2.1A - Input Lines from the Data bus to the CPU



Output Lines - The CPU sends data to the data bus through an octal buffer the DM74LS244 (no. 11 in the 4.3a). As with the input lines, this I.C. is activated through the read signal from the CPU. However, between the CPU and the I.C. (No.11) there is a hex inverter which inverts the RD signal to logic 1 when the CPU is writing, this signal then activates the I.C. (No. 11) and the data from the CPU is transmitted to the data bus.

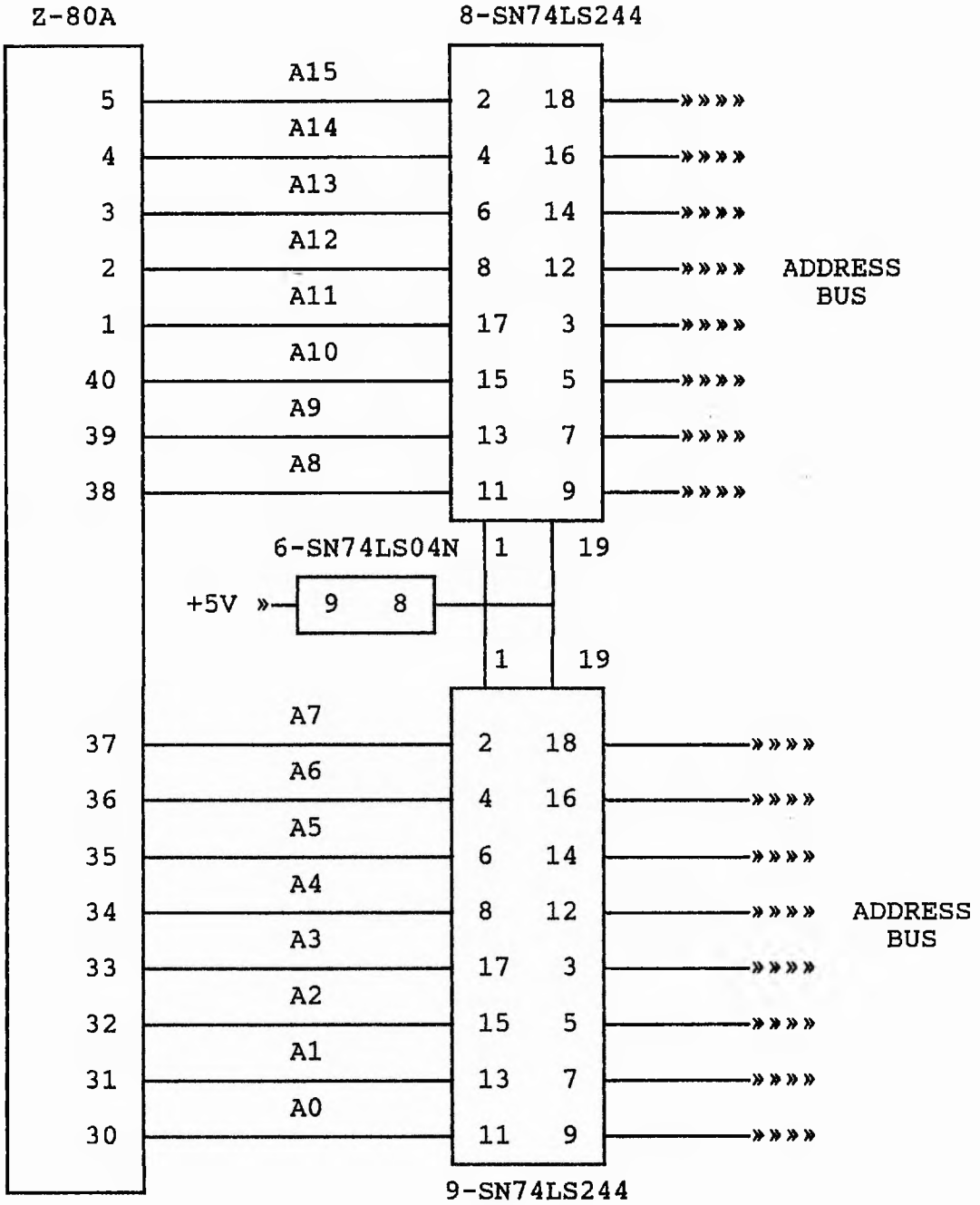
Figure 4.3.2.2.1B - Output lines from the Z-80A CPU



4.3.2.2.2 - Address Bus Lines

The CPU address lines are connected with the address bus of the system through two octal buffers DM74LS244 (See figure 4.3.2.2.2 I.C. nos. 8 and 9, respectively). These octal buffers are activated through a hex inverter (I.C. no. 6 in the 4.3.2.2.2) which is fed by +5 volts d.c producing logic 0 which activates these two octal buffers (see figure 4.3.2.2.2)

Figure 4.3.2.2.2 - Address Bus lines



4.3.2.2.3 - Logical Bus Lines

The logical bus lines of the PCS consist of 6 different lines as follow:

1 - $\overline{\text{MERQ}}$ (Memory Request) works together with the $\overline{\text{RD}}$ or the $\overline{\text{WR}}$ signal to read data from ($\overline{\text{MR}}$) or write ($\overline{\text{MW}}$) data into the memory, (figure 4.3.2.2.3a).

2 - $\overline{\text{IORQ}}$ (Input/Output Request) associated with $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signals, it can read or write data from or to external devices such as the Analog/Digital Converter (figure 4.3.2.2.3a)

3 - $\overline{\text{WAIT}}$ allows the CPU to be synchronised with the reading/writing speed of the memories, avoiding loss of data while a read or write cycle is in operation (figure 4.3.2.2.3b).

4 - $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (Read and Write) were explained in the above section (item 4.3.2.2.1).

5 - $\overline{\text{RESET}}$ initialises the CPU. This signal causes the Z-80A CPU to clear the PC register, and, after a short pause, start execution from memory location 0000H. Certain other CPU functions are also initiated when Reset becomes active(figure 4.3.2.2.3c).

The $\overline{\text{NMI}}$, $\overline{\text{INT}}$ and $\overline{\text{BUSREQ}}$ which are interrupt logic signals are not used in the circuit of the PCS. Figure 4.3.2.2.3a shows the logic bus lines used in PCS.

Figure 4.3.2.2.3a - Logic Bus Lines

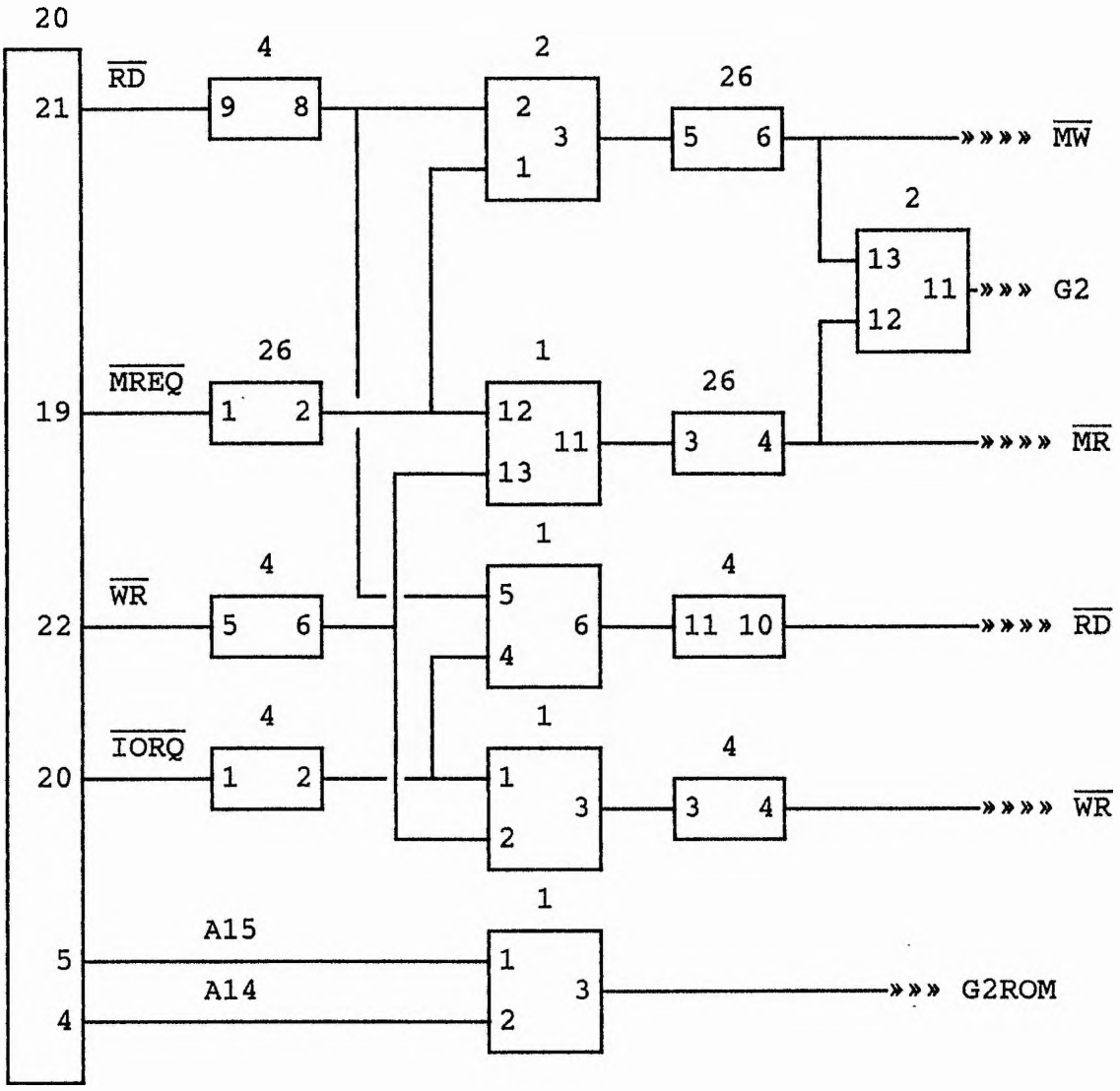


Figure - 4.3.2.2.3b - Diagram of the WAIT circuit in the PCS

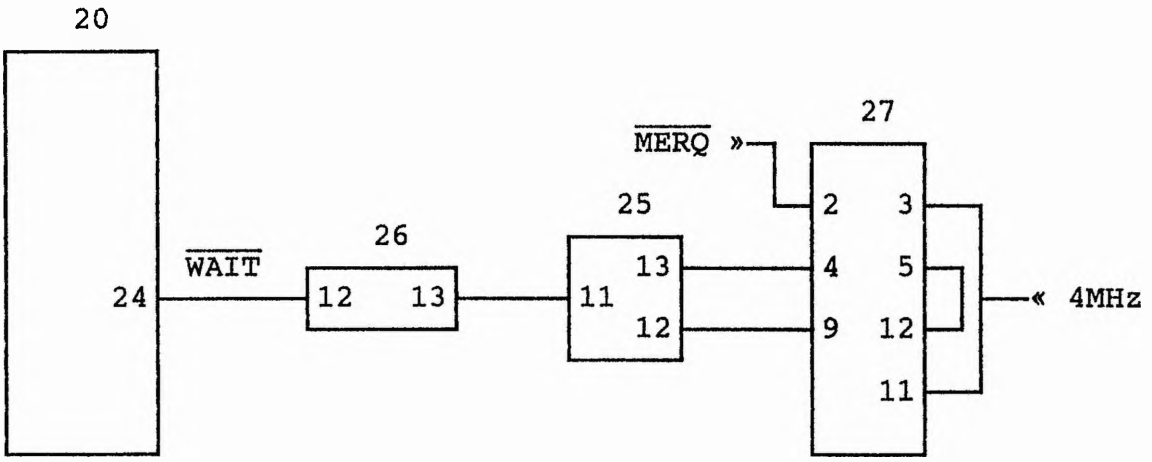
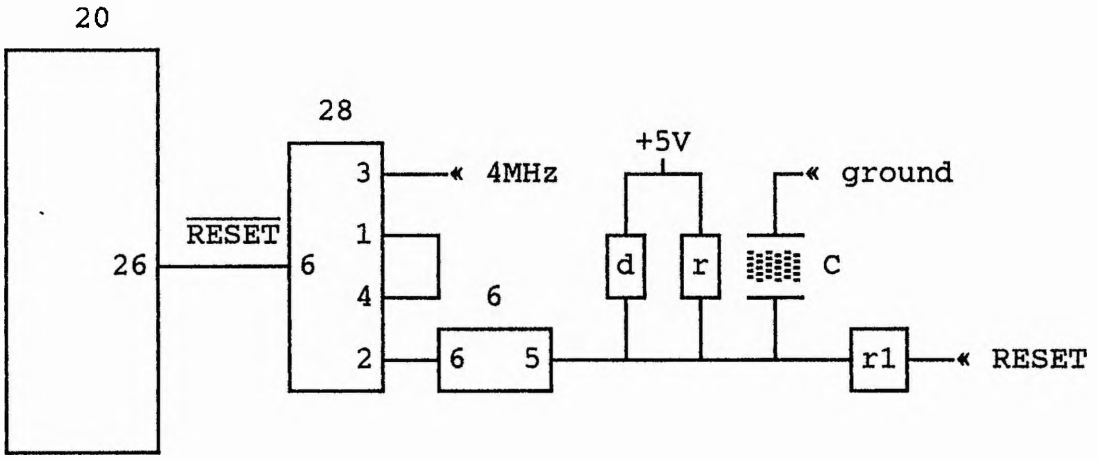


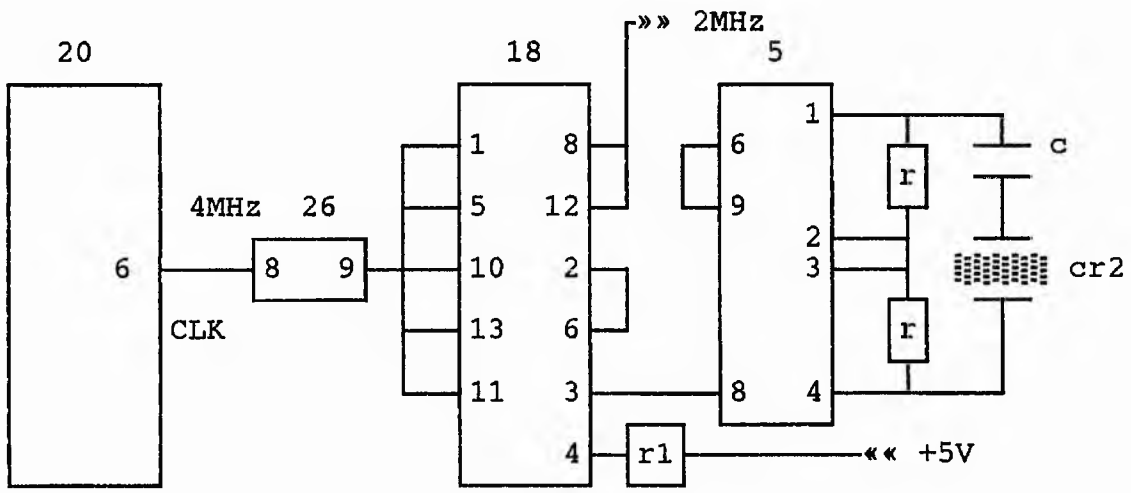
Figure 4.3.2.2.3c - Diagram of RESET circuit in the PCS



4.3.2.2.4 - The Clock Circuit.

The Z-80A CPU in the PCS circuit is clocked at a fixed 4MHz with a 250nsec cycle time. The clock signal is produced from a 8MHz quartz crystal (figure 1.1.2.D), which has had its frequency reduced to 4MHz and been synchronised by a circuit shown in figure 4.3.2.2.4.

Figure 4.3.2.2.4 - Diagram of the Clock circuit in the PCS

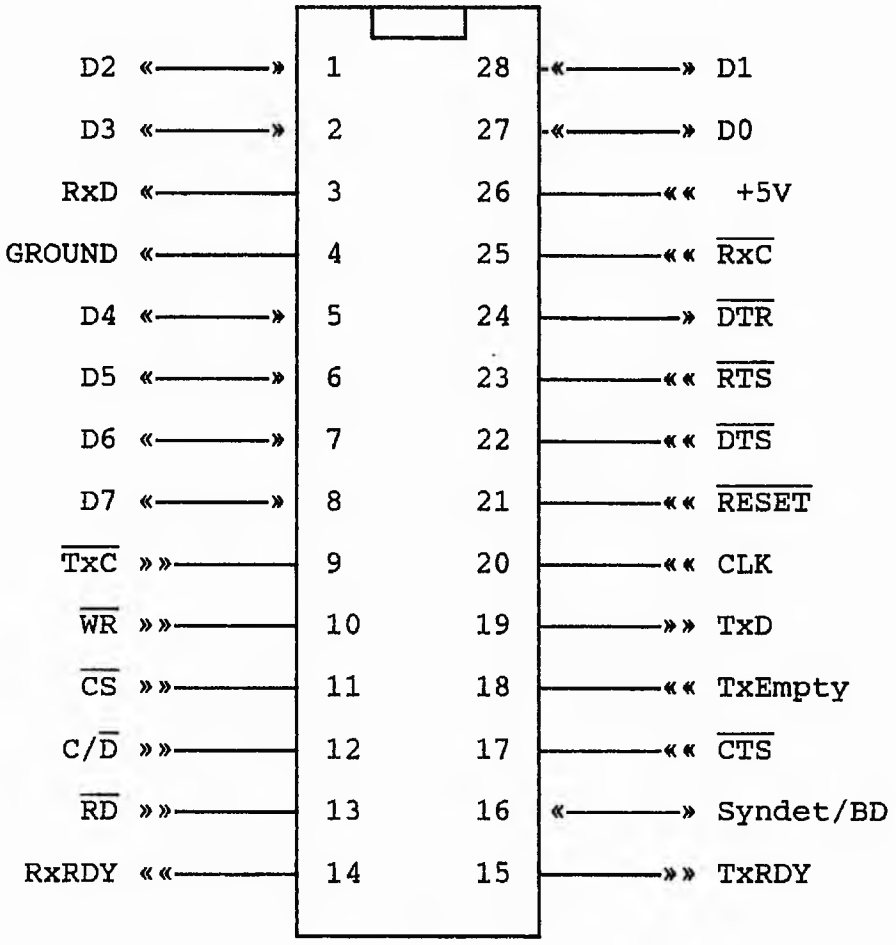


4.3.2.3 - Serial Input/Output Port Connections.

The PCS has two serial I/O ports. Each port has its own baud rate select switch enabling baud rates from 75 to 9600 baud (bit per second). Each port utilises the industry standard 8251A USART (Universal Synchronous Asynchronous Receiver/Transmitter).

The USART is an I.C. built in a 28-pin DIP package (figure 4.3.2.3) which accepts data characters from the CPU in parallel format and then converts them into continuous serial data streams for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. It works with a 2MHz clock rate, asssuming that all inputs and outputs are TTL compatible.

Figure 4.3.2.3 - Pin Configuration of the USART



Pin Names:

D0 - D7 Data Bus used for data exchange with the CPU

RESET - At logic 1 forces the USART into an 'Idle' mode. It remains at 'Idle' until a new set of control words is written into it to program its functional definitions.

CLK (Clock) - Used to generate internal device timing, it is pulsed with a 2MHz clock frequency from the clock circuit in the CPU board (see item 4.3.2.2.4 Clock Circuit).

$\overline{WR}$ (Write) - At logic 0 on this input informs the USART that the CPU is writing data or control words to the USART.

$\overline{RD}$ (Read) - At logic 0 on this input informs the USART that the CPU is reading data or status information from the USART.

C/$\overline{D}$ (Control/Data) - Used with the $\overline{WR}$ or $\overline{RD}$ inputs, controlling the word and also the status information.

$\overline{CS}$ (Chip Select) - At logic 0 selects the USART.

$\overline{DSR}$ (Data Set Ready) - Used to test modem conditions such as Data set ready.

$\overline{DTR}$ (Data Terminal Ready) - Used for modem control such as Data terminal ready.

$\overline{RTS}$ (Request To Send) - Used for modem control such as request to send.

$\overline{CTS}$ (Clear to Send) - At logic 0 the USART transmits serial data.

TxRDY (Transmitter Ready) - This output signals to the CPU that the transmitter is ready to accept a data character.

TxE (Transmitter Empty) - Is at logic 1 when the USART does not have any data to send.

$\overline{TxC}$ (Transmitter Clock) - Controls the rate at which the character is to be transmitted.

RxRDY (Receiver Ready) - Indicates that the USART contains a character that is ready to be input to the CPU.

$\overline{RxC}$ (Receiver Clock) - Controls the rate at which the character is to be received.

Syndet/BD - This pin is used in synchronous mode for syndet and may be used as either input or output, programmable through the control word.

Connections of USART1 (I.C. No. 13) and USART2 (I.C. No. 30) with different circuits on the CPU board are shown in figure 4.3.2.3A and 4.3.2.3B, respectively. They can communicate with an external device as a Video (VDU) Terminal or other microcomputer through the quad line receiver (DS1489 I.C. No. 22 on the CPU board), which converts the serial RS232C level from the VDU terminal or from another computer to a TTL serial USART compatible level. The quad line driver (DS1488 I.C. No. 21 in the CPU board) converts the TTL input logical level from the USART to RS232C output level to the VDU terminal or another microcomputer.

Figure 4.3.2.3a USART1 (I.C. No. 13) in the CPU-board.

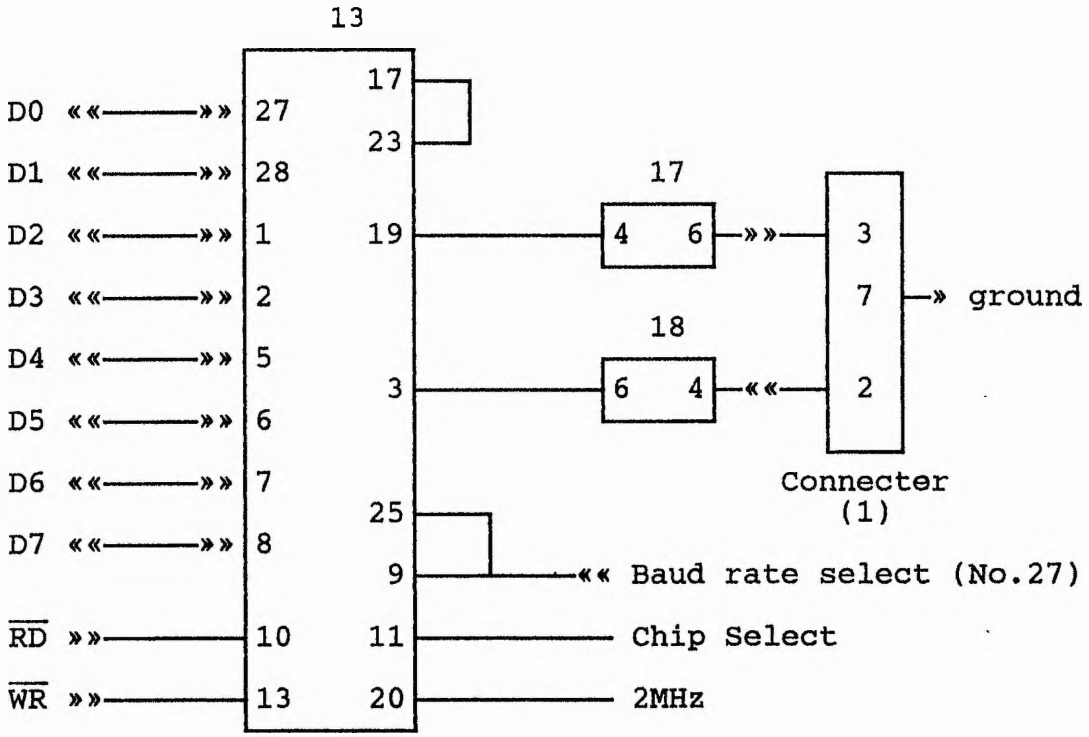
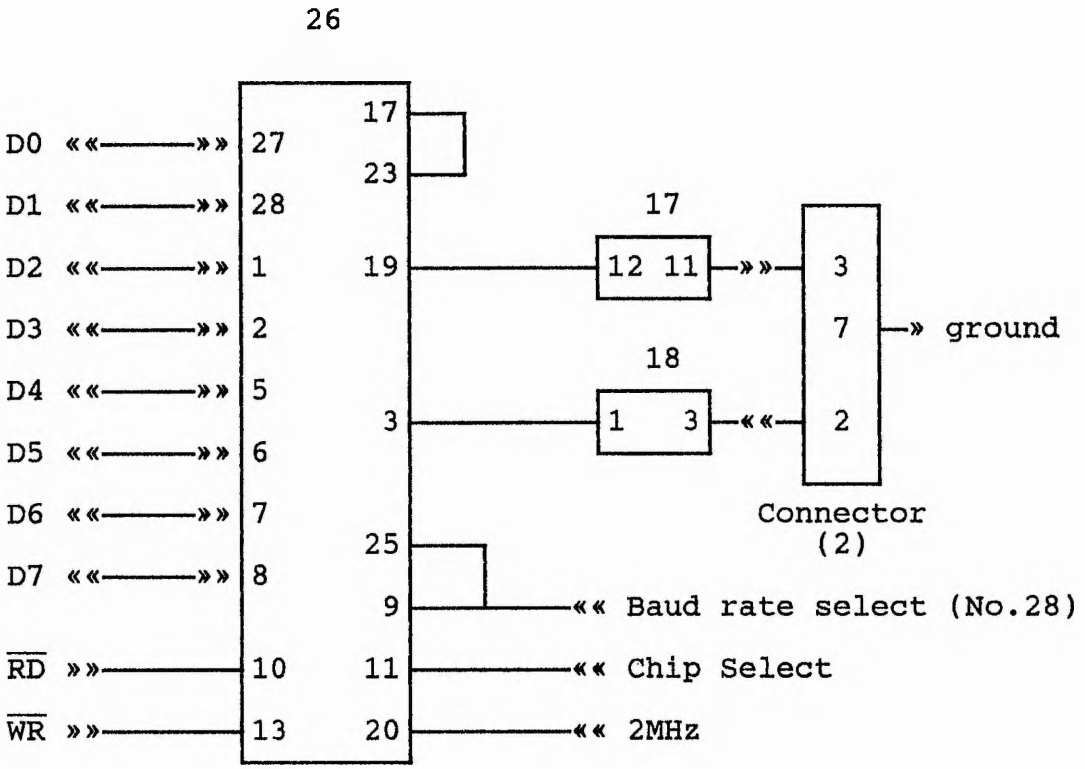


Figure 4.3.2.3B - USART2 (I.C. No. 30) in the CPU-board



4.3.2.3.1 - Programming USARTs (P8251A and D8251AC).

Prior to starting data transmission or reception, the USART must be loaded with a set of control words generated by the CPU. These control signals define functions of the USART and must immediately follow a reset operation (Internal or External).

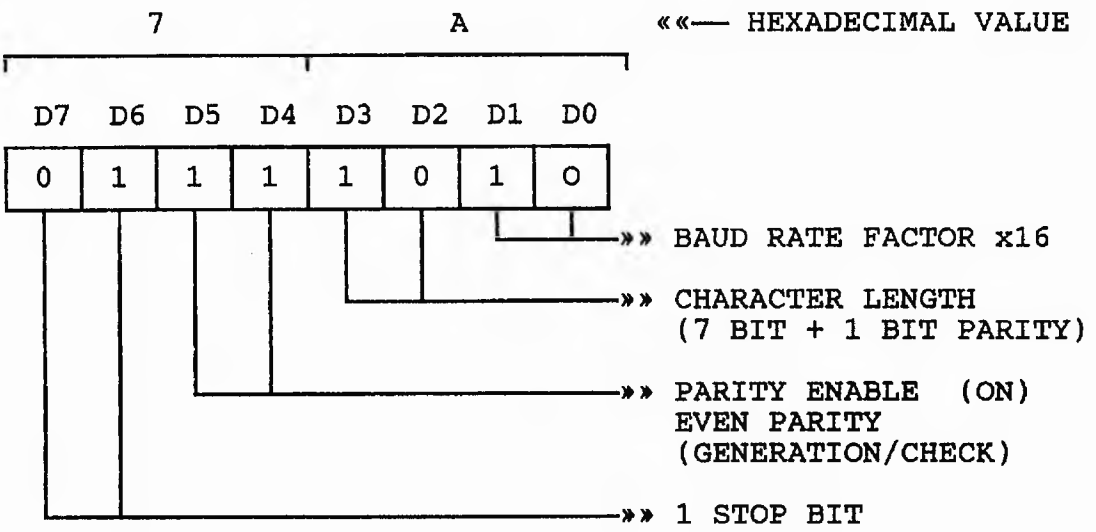
Control Words are divided into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction. Defines the general operational characteristics of the USART. Once the mode instruction has been written into the USART by the CPU, the Command Instruction may be inserted.

The USART can be used for either asynchronous or synchronous data communication. In figure 4.3.2.3.1A the mode instruction format of both USARTs used in this system (USART1 and USART2) is displayed.

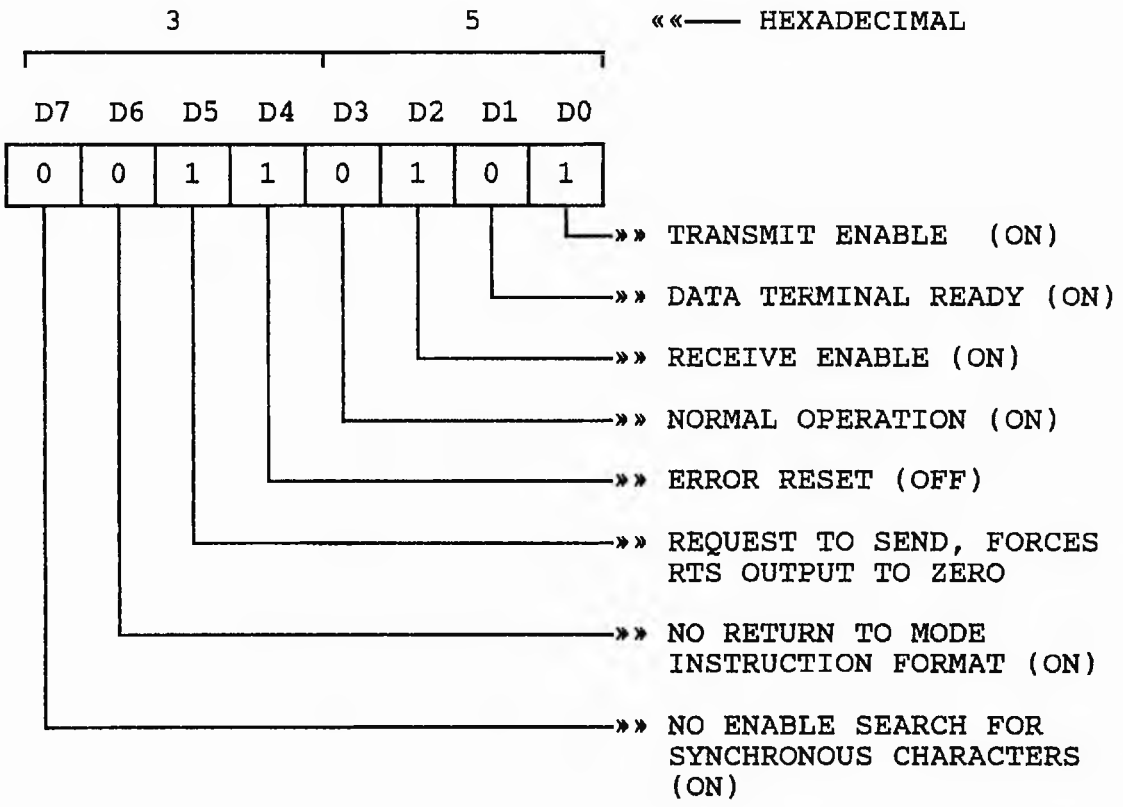
Figure 4.3.2.3.1A - Mode Instruction word



The mode instruction word used in this system is 7A (hexadecimal).

Command Instruction. Defines a status word that is used to control the actual operation of the USART. It controls the actual operation of the selected format defined in the mode instruction (see item - mode instruction -). In figure 4.3.2.3.1B the command instruction word format is described.

Figure 4.3.2.3.1B - Command Instruction word

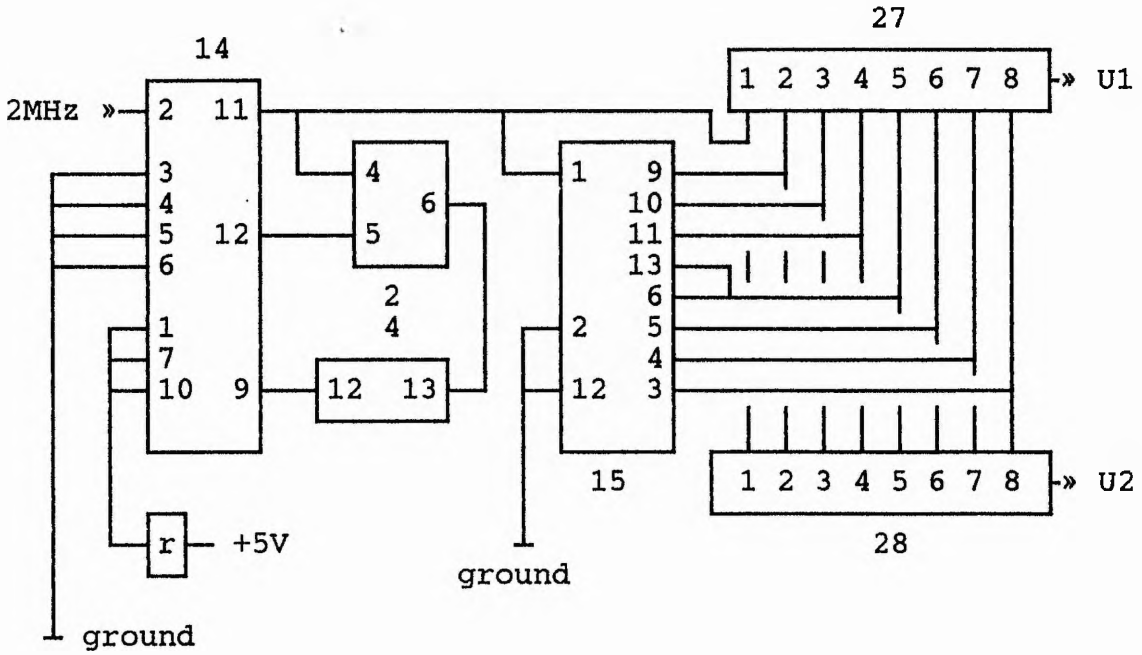


4.3.2.3.2 - Baud Rate System

The USARTs can transmit and/or receive data between 75 and 9600 bits per second. This baud rate is produced by the circuit described in figure 4.3.2.3.2 (below). One of these baud rates can be selected by means of a DIL in a Stackable switch (on/off), which corresponds to no. 27 in the

CPU board of a USART1 and no. 28 of the USART2.

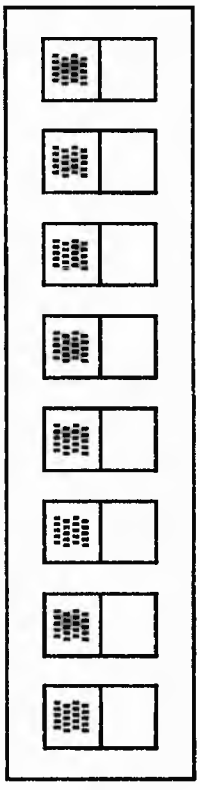
Figure 4.3.2.3.2 - Baud Rate circuit



U1 = USART1 U2 = USART2

DIL End Stackable Switch (on/off) configuration (27 and 28)

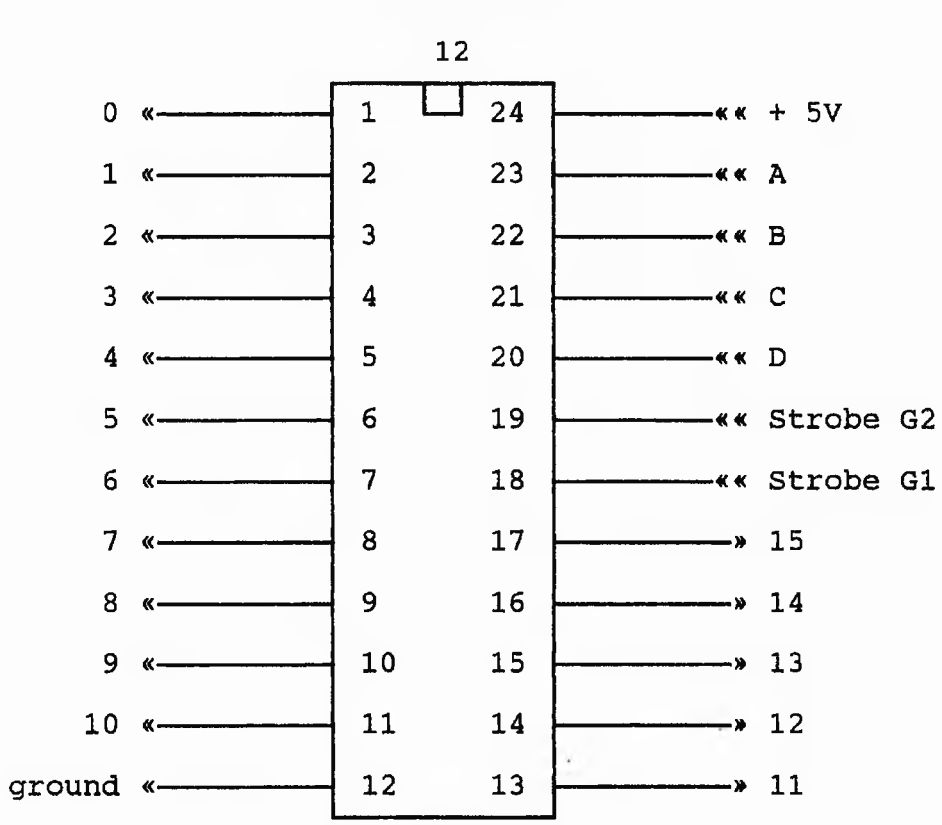
Pins	Baud Rate
1	75
2	150
3	300
4	600
5	1200
6	2400
7	4800
8	9600



4.3.2.4 - Demultiplexer SN74LS154N.

The PCS is able to control the access of several devices such as the RTC or the USARTs. For this a demultiplexer (the SN74LS154N ,I.C. No. 12 in the CPU board) is applied. It is capable of selecting sixteen devices using just four address lines from the CPU. Figure 4.3.2.4 shows the pin configuration of the Demultiplexer.

Figure 4.3.2.4 - Pin configuration of the Demultiplexer



Pin Names :

0 - 15 - Chip select to devices in the PCS.

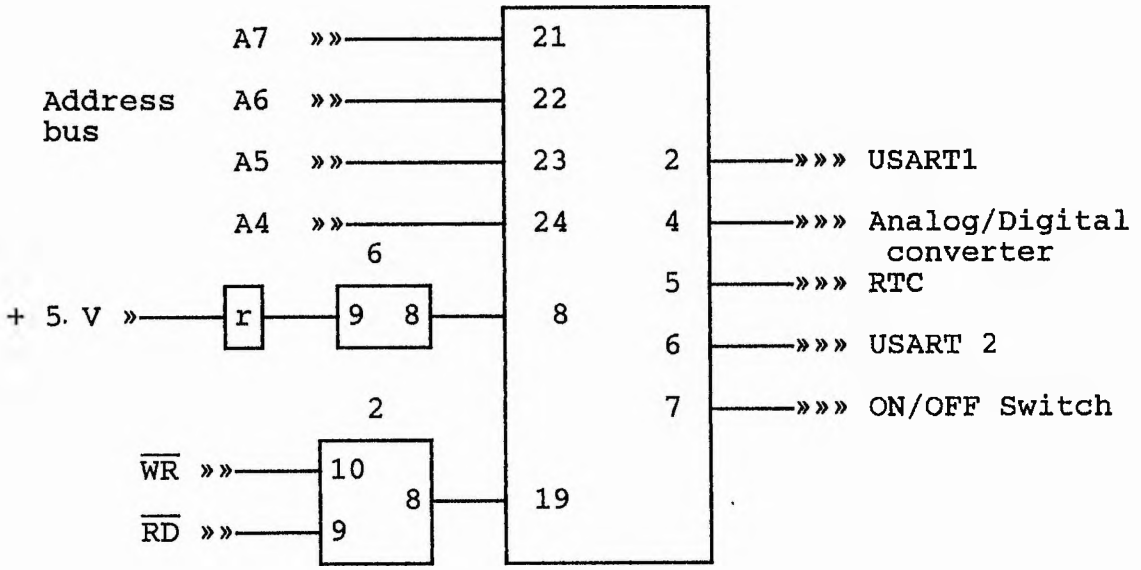
A - D - Input lines from the CPU.

Strobe G2 and G1 - Control lines to the Demultiplexer

4.3.2.4.1 - Demultiplexer connections.

The demultiplexer has a two line control which must be at logic 0 for it to work. The first control line (strobe G1) is connected directly with an output from an invertor (SN74LS04N, I.C. No. 6 in the CPU board) which maintains the strobe G1 continously at logic 0. The second control line is connected with an output from the 2-inputs AND gate (SN74LS08N, I.C. No. 2 in the CPU board). It is fed with the read and write control lines from the control bus, so when the CPU is reading or writing some information in the data bus the output of the AND gate will be at logic 0, too. The demultiplexer then becomes active to select one of the sixteen output lines whose address was written in the A7,A6,A5 and A4 bit of the address by the CPU. To find out which address corresponds, a specific output line is necessary to look at the truth table (table 4.3.2.4.1). The connections of the demultiplexer are described in figure 4.3.2.4.1.

Figure 4.3.2.4.1 - Connections of the demultiplexer in the CPU board



Components:

r - Resistor 1KΩ

6 - SN74LS04N

2 - SN74LS08N

Table 4.3.2.4.1 - Truth Table of the Demultiplexer SN74LS154N

INPUTS						OUTPUTS														D
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	1
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	A
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	R
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	2
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	S
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	

X ='not in use'

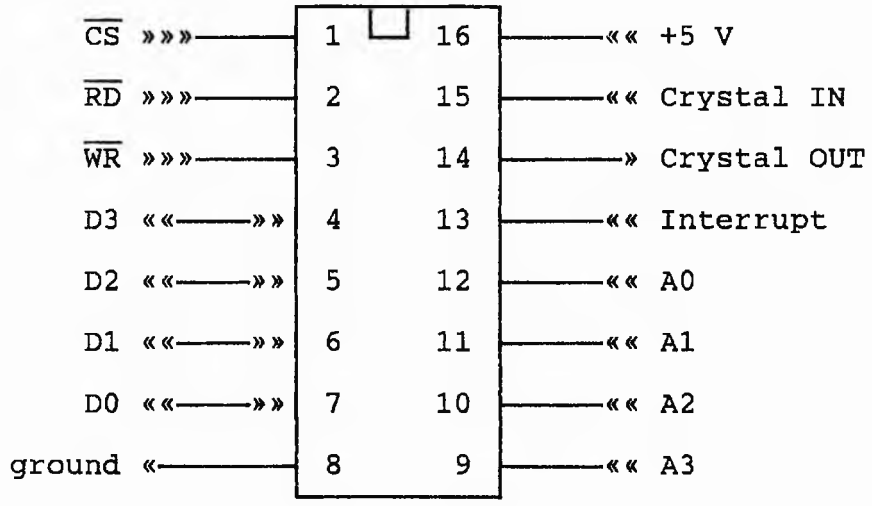
D = Device

Description of devices:
1 -» USART1
A -» Analog/Digital converter
R -» Real Time Clock
2 -» USART2
S -» ON/OFF Switch

4.3.2.5 - Real Time Clock Integrated Circuit (RTC)

The RTC (RS 58174, No. 19 in the CPU board) is a metal gate CMOS circuit that functions as a real time clock and calendar in bus-orientated microprocessor systems. It is packaged in a 16-pin DIP (Dual - in line package) and contains a 48-bit (14 digits) counter chain clocked from a 32,768 KHz crystal-reference oscillator. The RS 58174 is TTL-compatible and has low power standby operation and also can keep track of the time and communicate with the processor in any increment from 1/10 second to months. Figure 4.3.2.5 shows the pin configuration of the RTC.

Figure 4.3.2.5 - Pin Configuration of the RTC



Pin Names:

$\overline{CS}$ - Chip Select (From the Demultiplexer)

$\overline{RD}$ - Read $\overline{WR}$ - Write

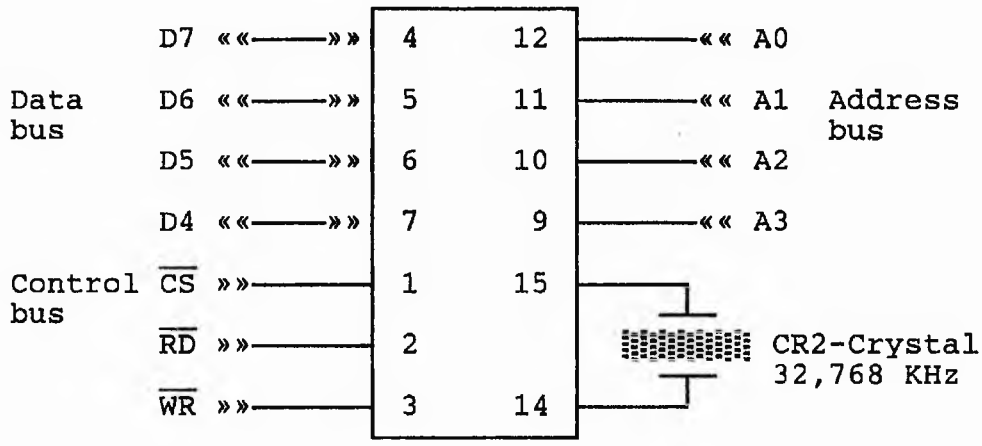
D0 - D7 - Data bus

A0 - A3 - Address bus

4.3.2.5.1 - Real Time Clock Connections.

The functions of the RTC in the CPU board are controlled by three control lines, four data lines, four address lines, the input and output signal from the crystal and an interrupt line. This circuit is described in figure 4.3.2.5.1.

Figure 4.3.2.5.1 - Connections of the RTC in the CPU Board



Control Lines:

1 - Chip Select ($\overline{CS}$) - This line is connected with the demultiplexer SN74LS154N (No. 12 in the CPU board), and it is used to activate the RTC.

2 - Write ($\overline{WR}$) - Together with the $\overline{CS}$ this command ($\overline{WR}$) permits the data from the data bus to be written in a suitable counter, whose address was fixed from the address bus.

3 - Read ($\overline{RD}$) - As above with $\overline{WR}$, $\overline{RD}$ works together with the $\overline{CS}$ command. When these control lines are activated at the same time, data from a specific counter are sent to the data bus of the system.

Data Lines:

The RTC has only 4 lines of data which are suitable to write and read values in a range from 0 decimal (00 hexadecimal or 0000 0000 binary) to up 15 decimal (0F - hexadecimal or 0000 1111 binary).

Address Lines:

Address lines consists of four lines, so the RTC has a word size of 4 bits with a range of 0 to 15 accessible counters.

Input/Output Crystal signal:

The Crystal, with a frequency of 32,768 KHz, provides, a pulse of 10 Hz which is used to increment all the seconds, minutes, hours, days, months and years counters. It is also associated with the data change flip-flop.

Interrupt Line:

Address 15 provides an interrupt output signal, which is a regular pulse selectable at intervals of 0.5, 5 or 60 seconds.

4.3.2.5.2 - Programming the Real Time Clock.

The Real Time Clock RS 58174 can be programmed to work with a processor in 2 different ways. The first way utilises the interrupt signal which can be programmed to produce a regulated signal at intervals of 0.5, 5 and 60 seconds. This pulse can stop the CPU with an interrupt command and every time that this occurs the counter in the memory is incremented. When the processor needs to know the real time, it must calculate this from the number of stops.

The second way, employed in the PCS, consists of taking readings from counters, then sending them using the four most significant digits in the data bus (see figure 4.3.2.5.1). For example, the tens of seconds counter (see table 4.3.2.5.2) increments its value by 1 each ten seconds going from 0 (0000 0000 binary) to 05 (0000 0101 binary), in 60 s then it is reset and starts again from 0. This counter was employed to "count" minutes. The Real Time Software will be further explained in the Software section.

For the RTC to start to operate normally it needs:

1 - To be reset, using the system reset;

2 - The values in the counters to be 0.

The sequence of events that initiates the operation of the RTC is described below:

1 - The value 00 (zero) is written in the 14th counter, (this stops the RTC);

2 - The value from the 4th to the 13th counters must also be zero;

3 - Logic 1 is written to the 14th counter (this switches on the RTC);

4 - The CPU is then able to take readings from a specific register (addresses in table 4.3.2.5.2.).

Table 4.3.2.5.2 - Address of the chip select and the counters of the RTC.

Counter	Address Hex.	mode
0 Test only	40	Write only
1 Tenths of sec.	41	Read only
2 Units of secs	42	Read only
3 Tens of secs.	43	Read only
4 Units of mins.	44	Read and write
5 Tens of mins.	45	Read and write
6 Units of hours	46	Read and write
7 Tens of hours	47	Read and write
8 Units of days	48	Read and write
9 Tens of days	49	Read and write
10 Day of week	4A	Read and write
11 Units of months	4B	Read and write
12 Tens of months	4C	Read and write
13 Years	4D	Write only
14 Stop/Start	4E	Write only
15 Interrupt and status	4F	Read and write

4.4 - The Memory Unit Board

Figure 4.4 shows the diagram of the memory board where the data from the CPU board are stored (see figure 4.4a). There are two kinds of memory in this system. Firstly, the Random Access Memory (RAM) in which the content of a memory cell (unit of the memory) can be erased through a software command or as a result of a power cut. This category of memory is extremely useful, since its content can be modified during program execution. Secondly, the Read Only Memory (ROM), is a type of memory in which the data is permanently fixed in memory cells and remain there even during a power cut. There is a special ROM called EPROM (Erasable Programmable Read Only Memory) where data can be erased through exposure to ultra violet (UV) light. This is possible because this category of memory has a window on the top which permits UV light to be directed onto the circuit in the memory package. Both types of memories (RAM and EPROM) are present in the memory board.

The CPU utilises both kinds of memory. EPROM memory has been used to store the master software which controls the operation of the microprocessor (see section 5. PCS - Software). However, for data storage from the fermenter sensors and also to store partial results of mathematical operations, the CPU utilises the RAM memory. In figure 4.4 the diagram of the memory board of the PCS is shown.

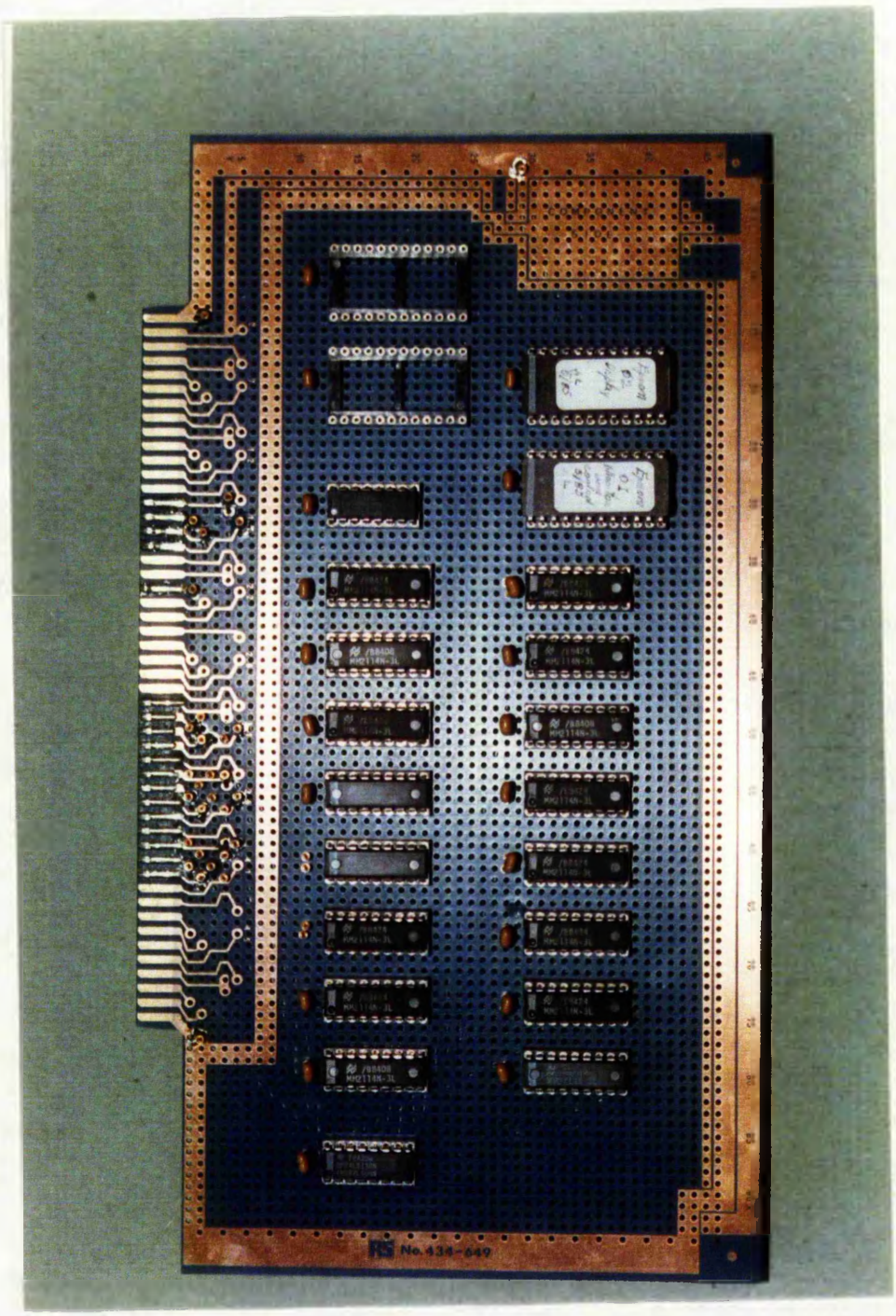
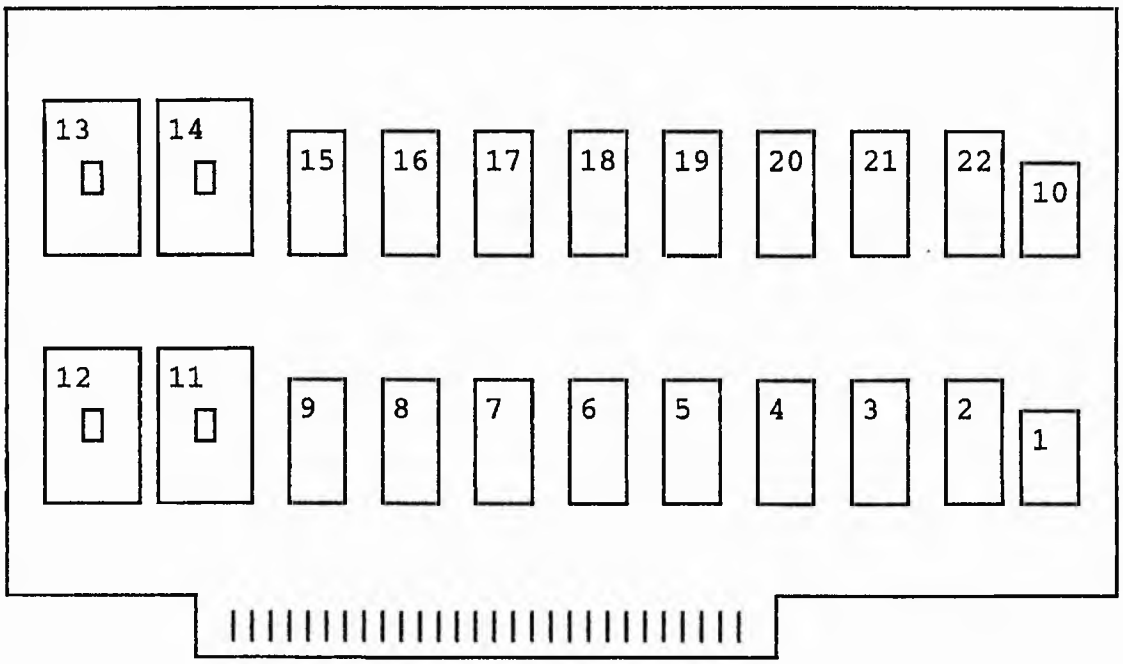


Figure 4.4a Photograph of the Memory Board of the PCS.

Figure 4.4 - Diagram of the Memory board of the PCS



Memory Board Components:

1 and 10 - DM74LS138N (3 to 8 Decoder/Demultiplexer) (N)

2 to 9 and 15 to 22 - MM2114N-3L (RAM) (N)

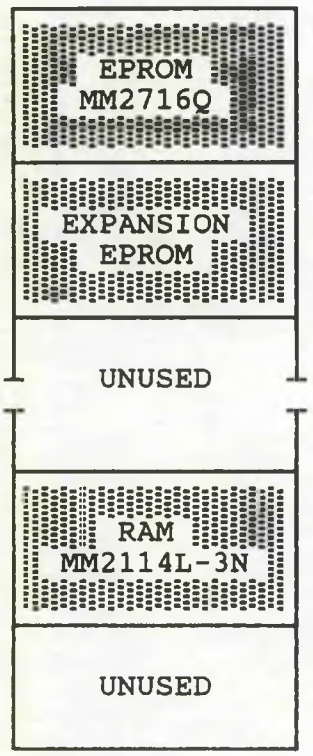
11 to 14 - MM2716Q (EPROM) (N)

(N) National Semiconductor Corporation (California, U.S.A)

4.4.1 - Addressing Memory

This memory system has a capacity of 16 Kbytes of binary information, where 8Kbytes are allocated for the RAM and the other 8Kbytes for the EPROM system. All space reserved for the RAM has been used, but only 4Kbytes of the EPROM have been utilised, (The master software of the system utilises less than 4 Kbytes). Figure 4.4.1 represents the map of the memory utilised in this system with its respective addresses.

Figure 4.4.1 - Memory map of the PCS



Address		
Decimal	Hexadecimal	
0	0000H	Master Program
2423	0977H	
2424	0978H	
8191	1FFFH	
8192	2000H	
49151	BFFFH	
49152	C000H	Auxiliary programs Data store
56319	DBFFH	
56320	DC00H	
65536	FFFFH	

4.4.2 - Principal circuits in the Memory Board.

The Memory board consists of three main circuits. The first is a set of two demultiplexers, of which the first has been used to select a specific group of two RAMs and the second to choose a specific EPROM. The second main circuit is the RAM circuit which consists of two groups of 8 chips (each chip with capacity to store 1 Kbyte with a word size of four bit (4x1024)); in other words the RAM circuit has space to store 8Kbytes of binary information distributed into 16 I.C., whereas the EPROM circuit is distributed into four I.C. (each I.C. is able to store 2Kbytes with a word size of 8 bit).

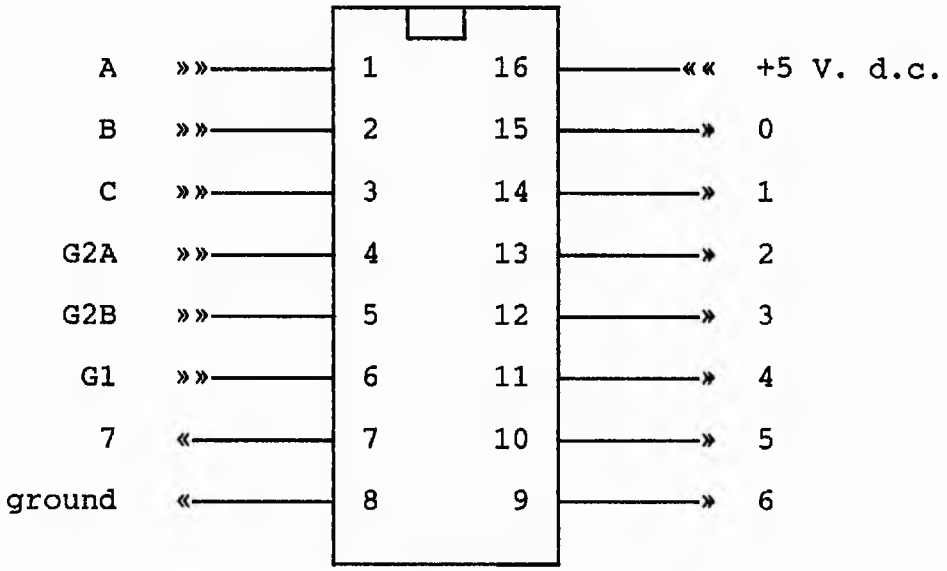
4.4.2.1 - The Demultiplexer Circuit.

The Demultiplexer (DM74LS138N) is a monolithilic complementary MOS (CMOS) I.C. (see figure 4.4.2.1). It has three input address lines which are connected with lines A10,A11 and A12 from the address bus of the system, and also has three strobe input lines of which two (G2A and G2B) must be at logic 0 and the third control input (the G1) must be at logic 1.

The function of the demultiplexer is totally controlled by software, which controls G2A and G2B, and of course selects one of eight output lines from a combination of the three address lines following the truth table described below (see table 4.4.2.1A for the RAM and table 4.4.2.1B for the EPROM system).

Figure 4.4.2.1 - Pin Configuration of the Demultiplexer

(SN74LS138N)



Pin Names:

A - C -» Address input lines

0 - 7 -» Chip select (output lines)

G2A,G2B and G1 -» Control lines

Table 4.4.2.1A - RAM truth table of the Demultiplexer

SN74LS138N

G2A	G2B	A	B	C	1	2	3	4	5	6	7	8	GROUP OF RAM	ADDRESS
L	L	L	L	L	L	H	H	H	H	H	H	H	5 AND 4	DC00-DFFF
L	L	L	L	H	H	L	H	H	H	H	H	H	2 AND 3	D800-DBFF
L	L	L	H	L	H	H	L	H	H	H	H	H	7 AND 6	D400-D7FF
L	L	L	H	H	H	H	H	L	H	H	H	H	8 AND 9	D000-D3FF
L	L	H	L	L	H	H	H	H	L	H	H	H	15 AND 16	CC00-CFFF
L	L	H	L	H	H	H	H	H	H	L	H	H	18 AND 17	C800-CFFF
L	L	H	H	L	H	H	H	H	H	H	L	H	19 AND 20	C400-C7FF
L	L	H	H	H	H	H	H	H	H	H	H	L	21 AND 22	C000-C3FF

Figure 4.4.2.1A describes the connections between the demultiplexer (I.C. No. 1 connected with the RAM system) and the bus system. The demultiplexer is controlled by two control lines, as follows :

1 - Address line A13 (logic 0) selects the RAM system from the EPROM one.
2 - The G2RAM line is always at logic 1 when the CPU is reading or writing.

Then the demultiplexer becomes active selecting a group of one of two I.C. RAMs since the G1 (pin 6) is constantly at logical value 1.

Figure 4.4.2.1A - Connections of the RAM demultiplexer in the Memory board

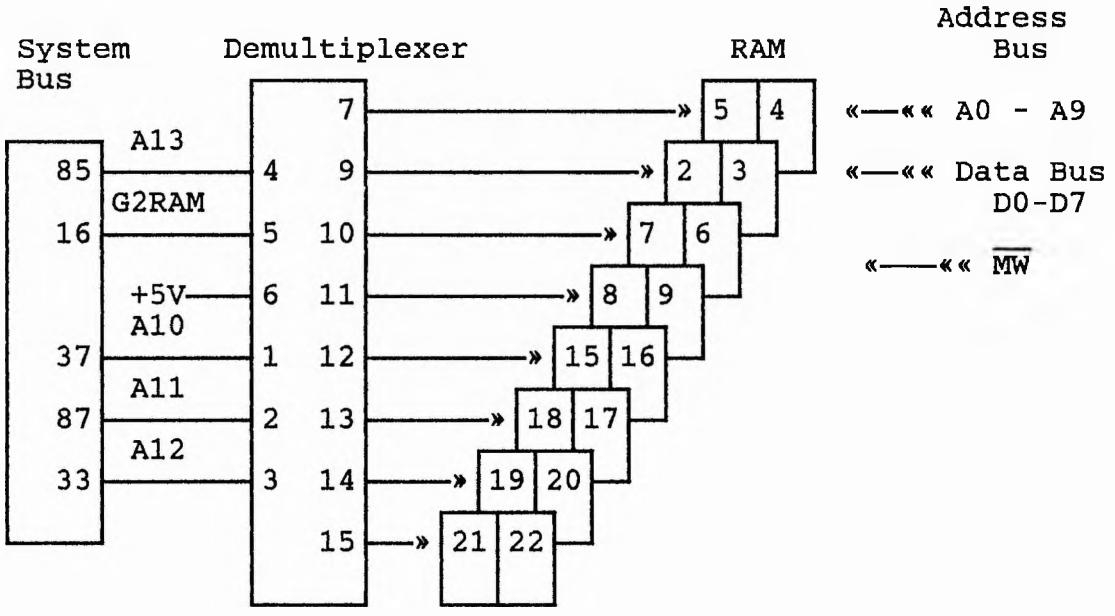


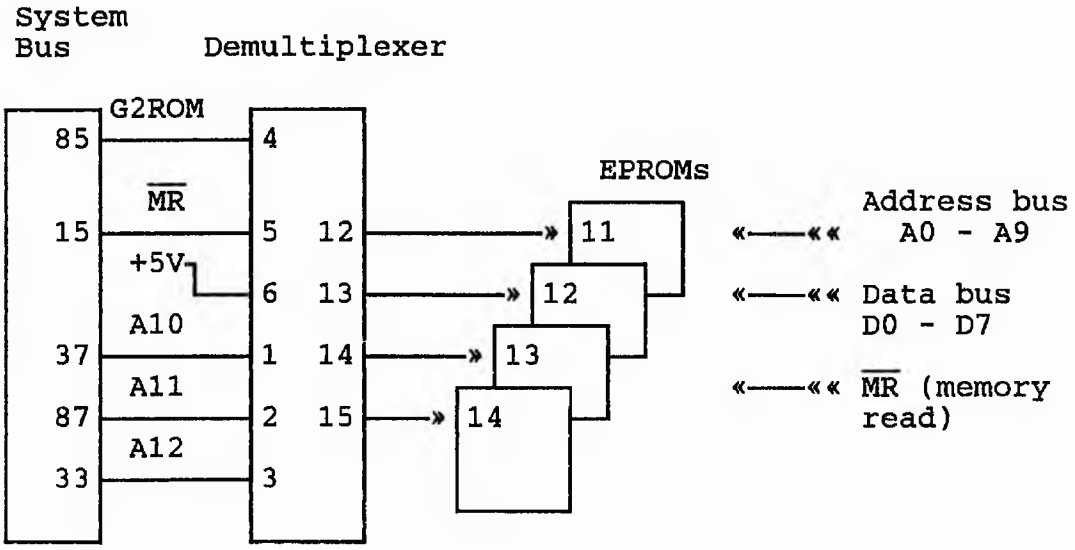
Figure 4.4.2.1B is the diagram of the EPROMs demultiplexer (I.C. No.10). The control line G2ROM (pin 4) is connected with the output of a 2-input OR gate which has its inputs linked with the address line A15 and A14, which are at logic 1, when the CPU is using the RAM system. But, following the addressing of the EPROMs in the truth table below (table 4.4.2.1B) when the CPU is using the EPROM system, these two control lines (G2ROM and MR) and the G1 Line (pin 6) must be at logic 0.

The other three lines A, B and C are connected with A11, A12 and A13 respectively from the address bus. These lines have been used to select a specific output line (chip select) in the demultiplexer using the truth table shown below (table 4.4.2.1B).

Table 4.4.2.1B -EPROM truth table of the Demultiplexer SN74LS138N

G2ROM	MR	A	B	C	1	2	3	4	5	6	7	8	EPROM	ADDRESS
L	L	L	L	L	L	H	H	H	H	H	H	H		
L	L	L	L	H	H	L	H	H	H	H	H	H		
L	L	L	H	L	H	H	L	H	H	H	H	H		
L	L	L	H	H	H	H	H	L	H	H	H	H		
L	L	H	L	L	H	H	H	H	L	H	H	H	12	1800-1FFF
L	L	H	L	H	H	H	H	H	H	L	H	H	13	1000-17FF
L	L	H	H	L	H	H	H	H	H	H	L	H	14	0800-0FFF
L	L	H	H	H	H	H	H	H	H	H	H	L	15	0000-07FF

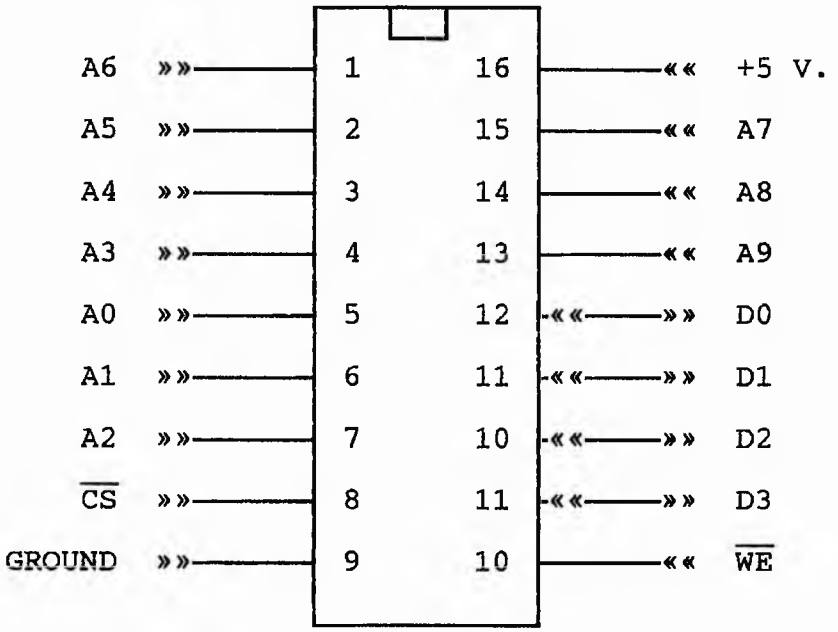
Figure 4.4.2.1B - EPROM demultiplexer connections in the memory board



4.4.2.2 - The Random Access Memory (RAM) (MM 2114L-3N)

A 18-pin static RAM is organised as 1024 words of 4 bits each with an access read time of 300 ns. They are called static because they can hold information indefinitely as long as they have power and the cell memory contents remain unchanged once set. In figure 4.4.2.2 the pin configuration of this memory I.C. is represented.

Figure 4.4.2.2 - Pin configuration of the RAM MM2114L-3N



Pin Names:

A0 - A9 -» Address bus

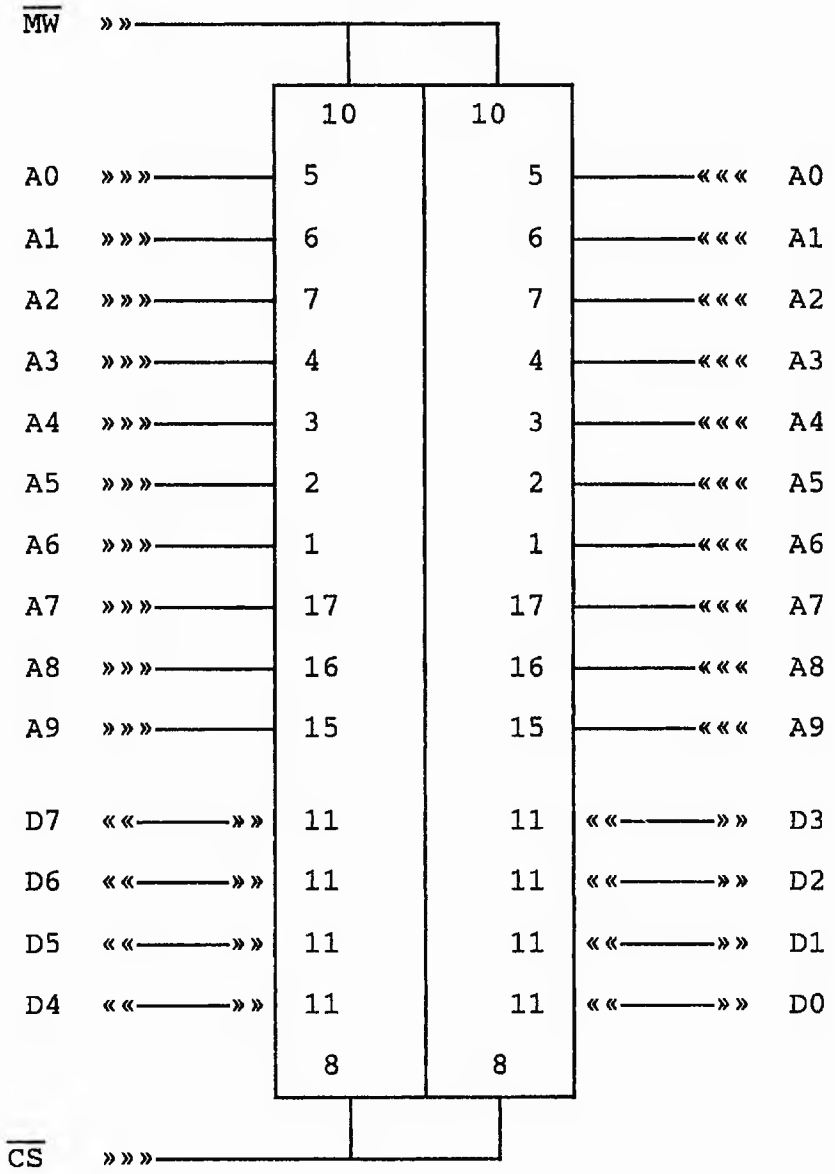
D0 - D4 -» Data bus

$\overline{CS}$ -» Chip select

$\overline{WE}$ -» Write Enable

Figure 4.4.2.2A (below) shows the connections of RAM memories with the rest of the system. Each two I.Cs. (MM2114L-3N) are using the same address bus lines and the same chip select line from the demultiplexer, and also the same Write Enable (WR) line which is connected with the memory write (MW) line from the CPU board. The CPU board selects the read mode of the RAM when is at logic 1 and the write mode when is at logic 0, but the data bus lines are different in the group of two RAMs. The four less significant bits go to one RAM and the most significant bits go to another one, because each RAM is able to store only four bits of one word of two in one Kbyte (4X1024).

Figure 4.4.2.2A - Connections of two RAM in the memory board



A0 - A9 -» Address bus lines

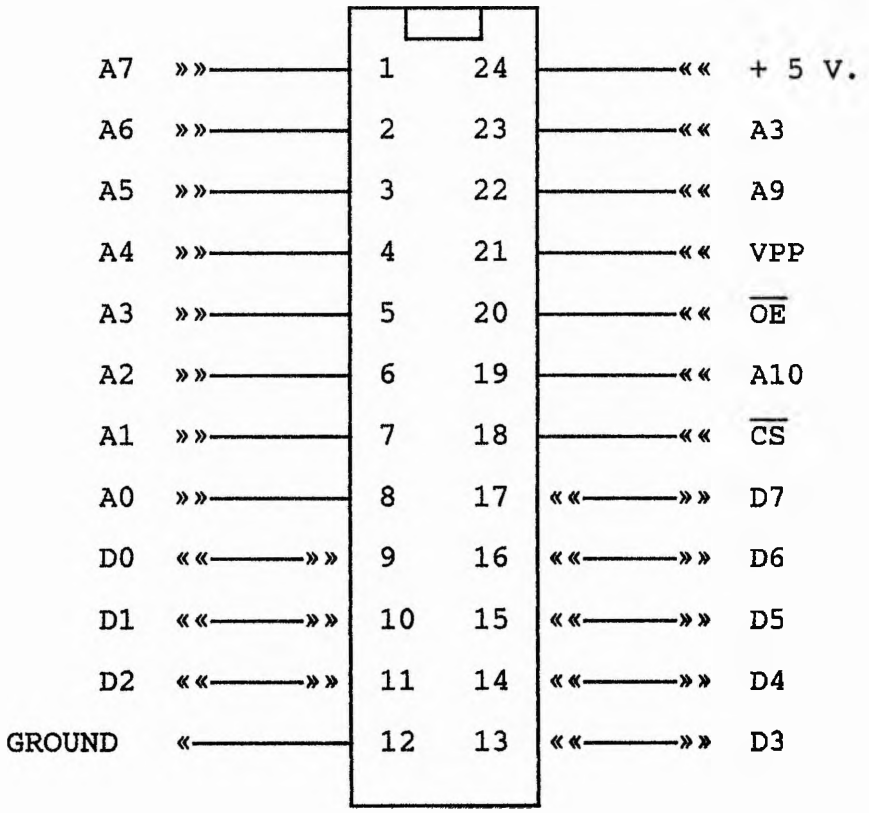
D7 - D4 -» Data bus (most significant bits)

D3 - D0 -» Data bus (less significant bits)

4.4.2.3 - The Erasable Programmable Read Only Memory (EPROM - MM 2716Q)

The MM2716Q is an ultra-violet light-erasable, electrically programmable read only memory. It has 16,384 bits organised as 2048 words of 8-bit length. The device is made using N-channel MOS silicon-gate technology for high speed and simple interfacing with MOS or bipolar circuits. It has an access time of 450ns. The MM2716Q is packaged in a 24-pin (see figure 4.4.2.3 - Pin configuration) dual-in-line package with transparent lid, which permits the chip to be exposed to ultra-violet to erase its bit pattern. Normally 30 mins exposure of the memory circuit to a violet light of 2537 Å wavelength yielding a total integrated dosage of 15 watt-second/cm^2 is required.

Figure 4.4.2.3 - Pin Configuration of the EPROM MM2617Q



Pin Names:

A0 - A10 -» Address bus inputs

D0 - D7 -» Data bus

$\overline{CS}$ -» Chip select

$\overline{OE}$ -» Output enable

VPP -» With 5 V the data present in the memory cell can be only read. On the other hand, if the voltage is raised to 25 V then data can be stored via the data bus.

Figure 4.4.2.3A (below) describes the connections of the EPROM MM2716Q with other components of the memory board and also with the rest of the system. It has basically four sets of connections.

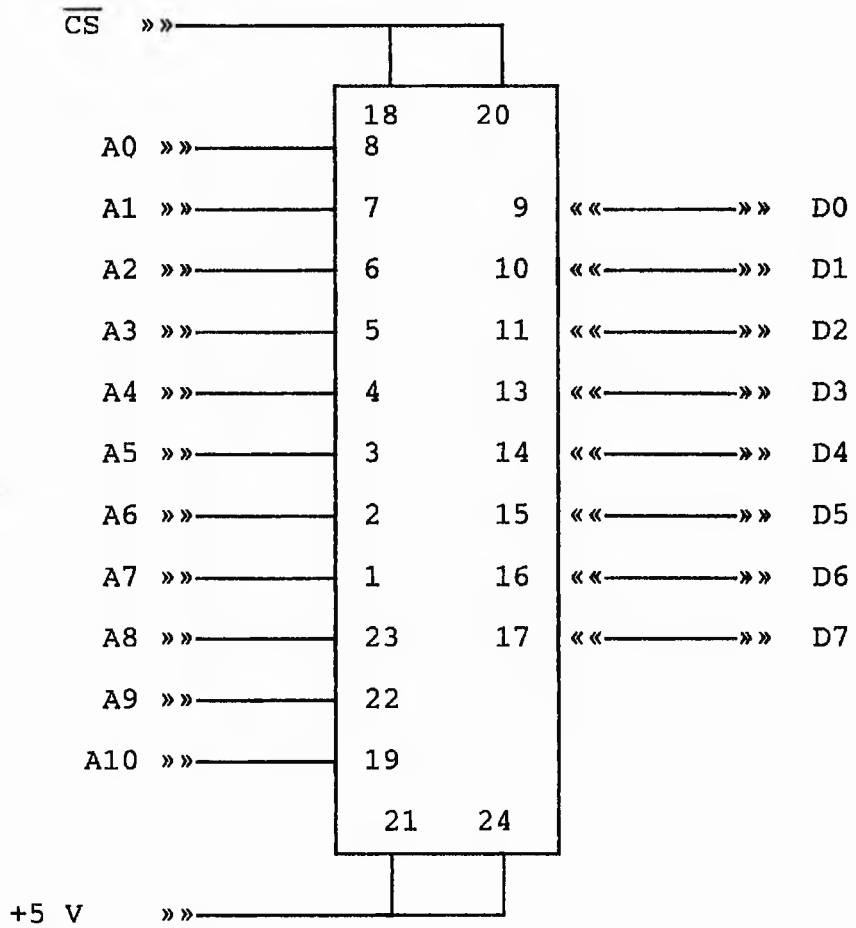
1 - Pins 20 and 18 are connected together with the chip select line from the demultiplexer (No. 10).

2 - Pins 21 and 24 are linked directly to a +5 V source which permits the memory to be read, but not be written.

3 - D0-D7 data bus system. Each memory I.C. is capable of storing a word of eight bits in 2Kbytes (8x2048)

4 - A0-A10 address bus.

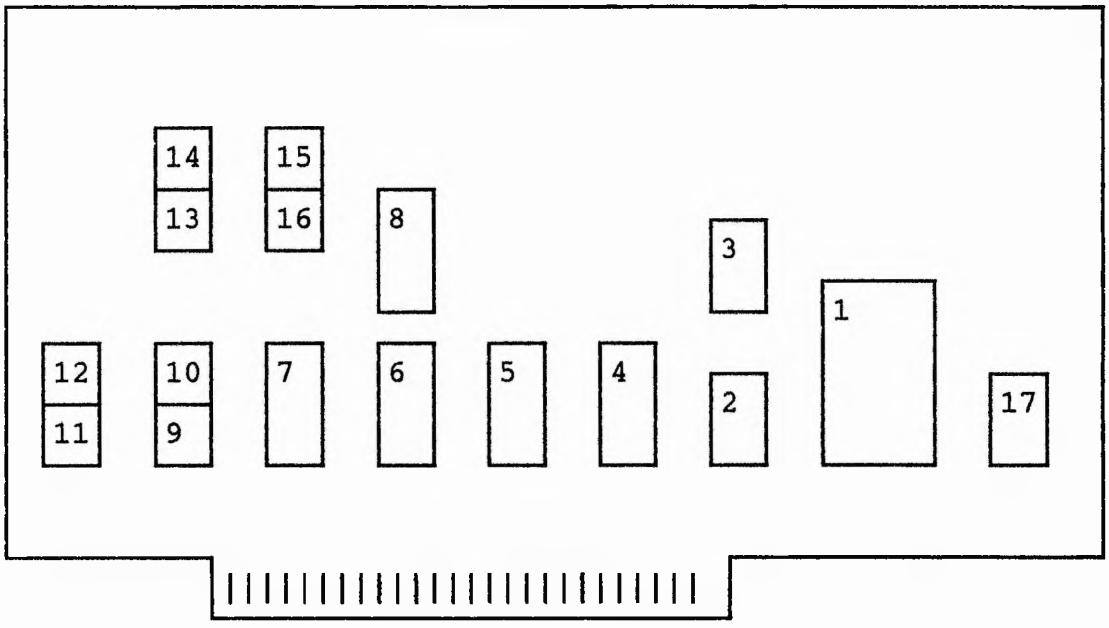
Figure 4.4.2.3A - Connections of the EPROM MM2716Q in the Memory Board



4.5 - Analog/Digital Converter (A/D) and ON/OFF Switch Board (ADS board)

The connection between fermenter and the microcomputer system is through an the Analog/Digital (A/D) converter and ON/OFF Switch board (ADS), which converts analog data from the fermenter sensors to digital data, before sending them to the CPU for processing and comparison with set-point data stored in memory. If any parameter value needs adjustment, this will be effected through the ON/OFF Switch circuit which can switch ON or OFF a specific device used to control a specific parameter inside the fermenter vessel such as the heater to control the temperature or the air pump to control the oxygen level. Figure 4.5 below shows the diagram of this board.

Figure 4.5 - Diagram of the Analog/Digital Converter and ON/OFF Switch Board.



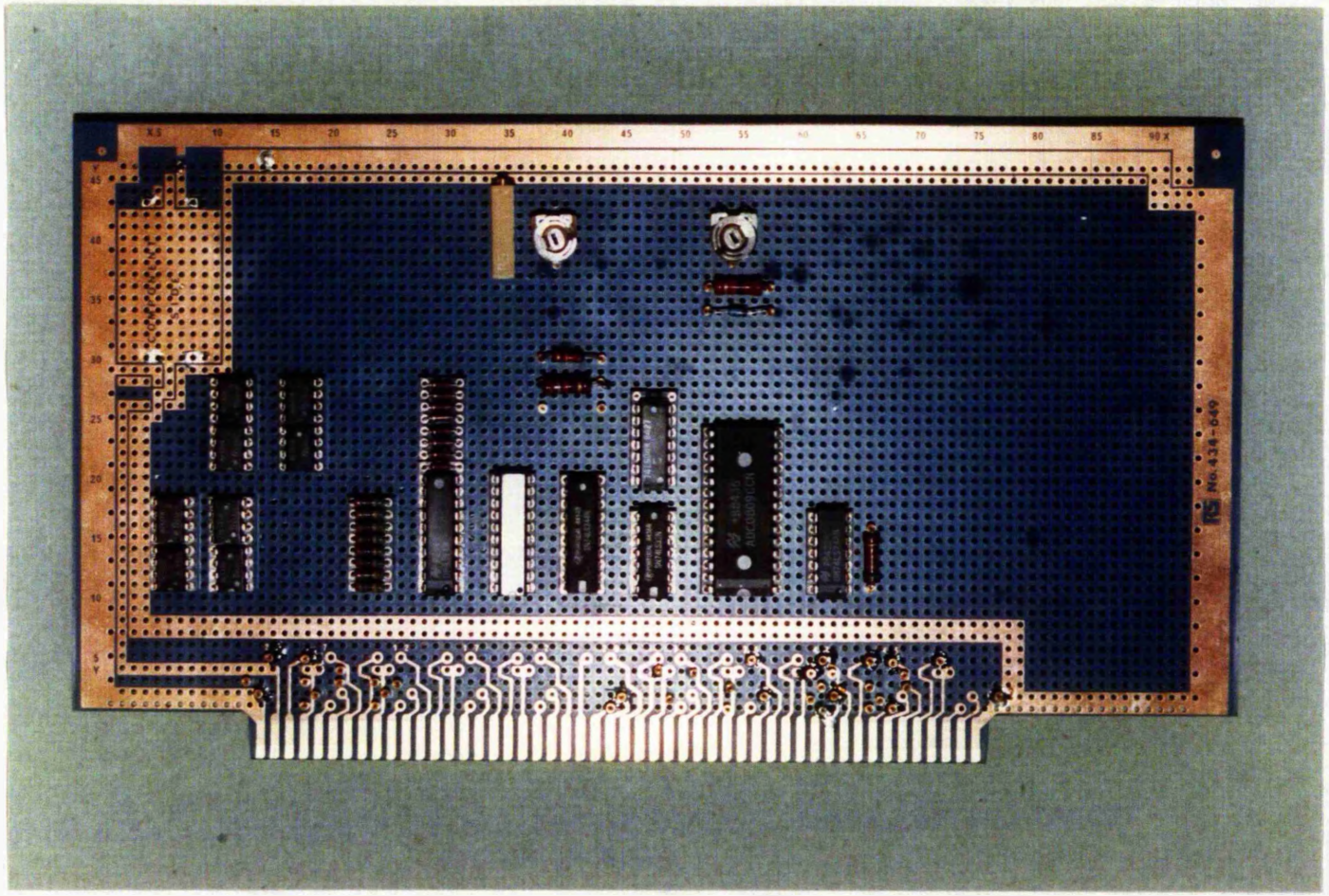


Figure 4.5a - Photograph of the Analog/Digital Converter and ON/OFF Switch Board.

Components:

1 - ADC0809CCN (Microcomputer compatible A/D Converter with 8-channel multiplexer (N)

2 - DM74LS32N (Quad 2-Input OR Gate).(N)

3 - DM74LS04N (Hex Inverter).(N)

4 - SN74LS244N (Octal Buffer and Line Driver with Tri-state outputs) (T).

5 - DM74LS244N (Octal Buffer and Line Driver with Tri-state outputs) (N).

6 - DP8311N (Octal Latched Peripheral Driver) (N)

7 - Set of 8 resistors (1.2KΩ, 0.125W) (RS)

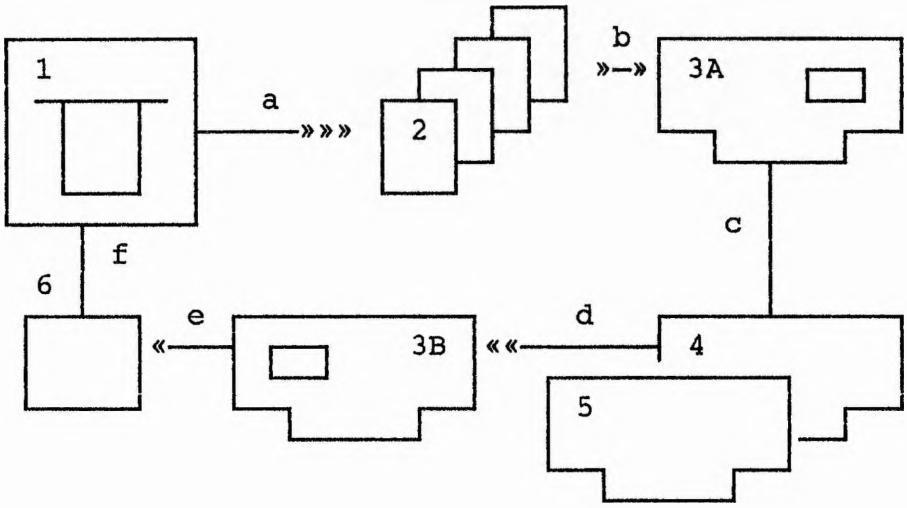
8 - Set of 8 resistors (1.1kΩ, 0.125W) (RS)

9 to 16 - 741CN (Operational Amplifier) (N)

17 - DM74LS74AN (Dual D Flip-Flop) (N)

In the figure 4.5 below is the block diagram of this board, which helps us to understand its function.

Figure 4.5 - block diagram of the A/DS board.



Components: 1 - The Fermenter vessel

2 - Interface boards

3A- A/D circuit on the ADS board.

4 - CPU Board.

5 - Memory Board

3B- ON/OFF Switch circuit on the ADS board.

6 - Box control.

a - Analog signals from sensors go to interface boards, where their output will be scaled to a range between 0 and 2.55 volts;

b - The output from the interface boards will be transformed from analog to digital using the A/D converter (ADS) on the ADS board;

c - Digital signals will be sent to the CPU and memory board for analysis, storage and control;

d - If any parameter value requires adjustment the CPU will sent a command to the ON/OFF switch board (3B) (and hence, through the control box, to) to effect this

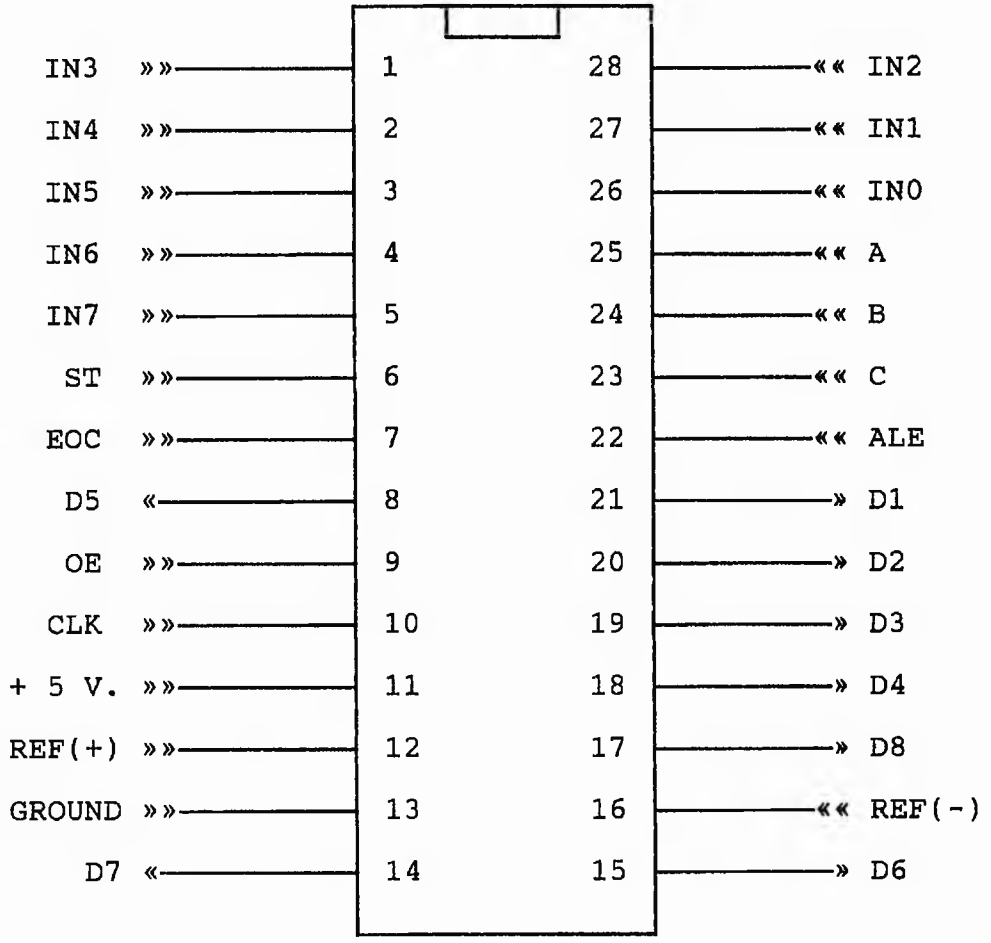
adjustment.

4.5.1 - Principal circuits in the ADS board.

4.5.1.1 - The Analog/Digital converter.

The ADC0890CCN is a monolithic complementary MOS (CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. It has a conversion time of 100µ seconds and no zero or full-scale adjust is required. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter shows a high impedance slitter stabilized comparator, a 256R voltage divider with a analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals. Figure 4.5.1.1 shows the diagram of this I.C. and the connections of the A/D converter with the PCS are described in figure 4.5.1.1A.

Figure 4.5.1.1 - Pin Configuration of the ADC0809CCN



Pin Names :

IN0-to-IN7 -» Analog signal inputs.

ST -» Start signal (ON = logical 1)

REF(+) -» Positive reference for the comparator (Volts)

REF(-) -» Negative reference for the comparator (volts)

CLK -» Clock signal (500KHz)

EOC -» End-of-conversion. At the end of a conversion the I.C. produces a logic 1, which may well be useful to control the operation of the CPU with this I.C., using a interrupt signal.

OE -» Output enable. Logic 0 during the conversion.

D0-to-D7 -» parallel data to the data bus.

A,B and C -» Input lines to select one of the eight input lines of the multiplexer.

ALE -» Address latch enable.

One of the eight analog channels will be selected following the truth table 4.5.1.1.

Table 4.5.1.1 - Truth table of the ADC0808CCN

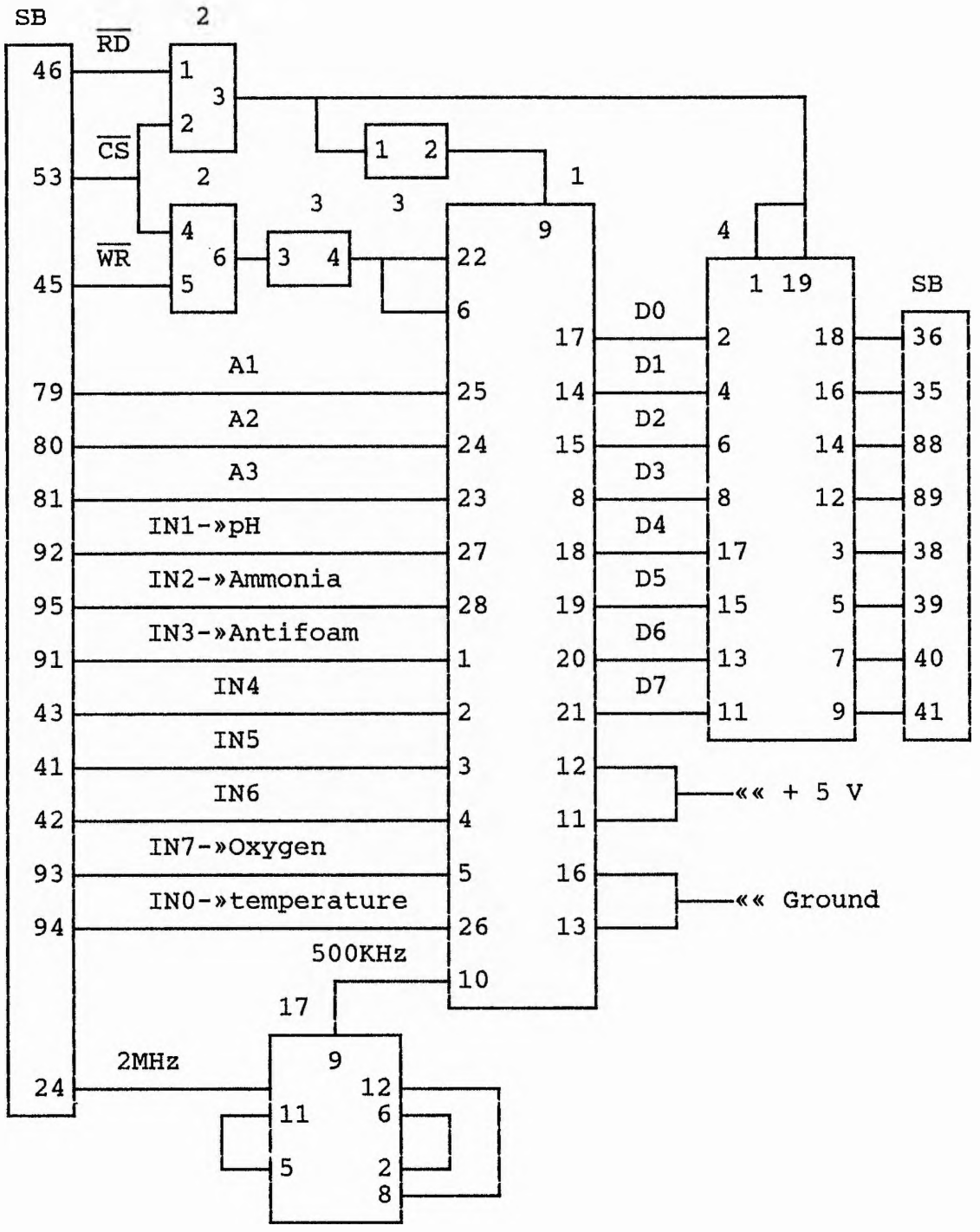
Address line			Selected Analog Channel		Fermenter parameters
A	B	C	NO.	Address	
0	0	0	IN0	30	Temperature
1	0	0	IN1	31	Ammonium system
0	1	0	IN2	32	pH
1	1	0	IN3	33	
0	0	1	IN4	34	Antifoam
1	0	1	IN5	35	Oxygen
0	1	1	IN6	36	
1	1	1	IN7	37	

As you can see in the table above the converter has the capacity to select eight channels, but in the PCS only five channels have been used. The address data are divided into two parts:

1 - The number 3 (011) is used to select the converter through the demultiplexer (SN74LS154N);

2 - The following numbers from 0 to 7 are used to select a specific channel using the internal multiplexer in the converter.

Figure 4.5.1.1A - Connections of the ADC0809CCN with the rest of the board and with the PCS



SB -» System Bus

The Converter has a timing clock of 500KHz derived from the 2MHz line of the system bus. It is selected through a chip select line from the demultiplexer (SN74LS154N) in the CPU board (see the CPU board and the Demultiplexer truth table for further information).

The converter has three control lines to control its function. The ALE and START lines are activated from a chip select signal (from the demultiplexer in the CPU board) and a write signal which is also from the control bus . Another line is the Output enable which is derived from the chip select line and the read line from the control bus of the system.

The A/D has two reference lines, one for a positive (REF(+)) and another for negative reference (REF(-)). These two references lines are used to produce a scale of conversion between 0 and 255 decimal (0000 0000 to 1111 1111 hexadecimal) which corresponds to the minimum and maximal values the converter can handle. In other words, if a specific device has a output range of 0 (minimum) to 5 v d.c (maximum), when it is connected with one of the inputs lines of the converter the output value in binary will be 0000 000 up to 1111 1111.

The converter functions are two-fold:

1 - To convert analog signals to digital ones.

2 - To send digital data to thedata bus of the system.

The conversion begins when the CPU starts to read data from one specific device which is connected with the converter. At this moment two control lines are activated the Chip Select (CS) which selects the converter and a specific channel and a write (WR) line (see figure 4.5.1.1A).

After a period of 8 clock pulses (the time which the converter needs to convert the analog signal to a digital one), a read (control line) will be activated by the CPU then the converter puts the 8-bit word on the data bus. However, to avoid problems of data transfer between the converter with data from other parts of the PCS, a buffer driver (I.C. 4 in the diagram above) was set up between the output data lines from the converter and the data bus of the system.

4.5.1.2 - The ON/OFF Switch.

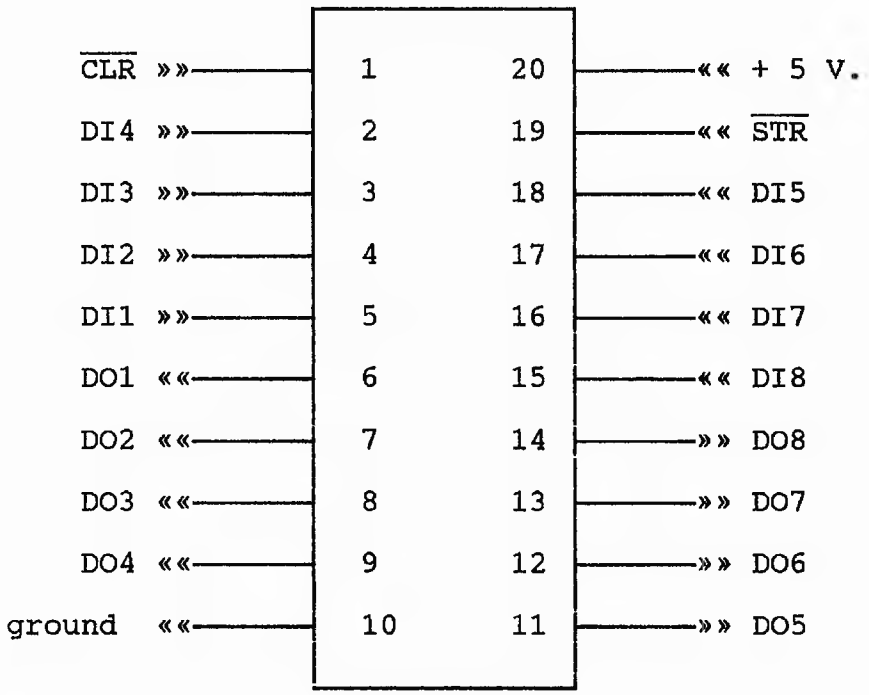
The ON/OFF Switch circuit controls the control box which controls certain parameters inside the fermenter vessel using specific devices (see table 4.5.1.2).

Table 4.5.1.2 - Parameters controlled by the ON/OFF switch and the Control Box.

Parameter	Device
Temperature	heater
Dissolved oxygen	air pump
pH	two peristaltic pumps
Foam	peristaltic pump

This circuit which is shown in the figure 4.5.1.2A has a octal latched peripheral driver (DP8311N) as the main I.C., which can latch eight bits of data with open collecter outputs. This I.C has been used to switch ON or OFF a specific device in the CONTROL BOX (see figure 4.5.1.2A - Pin configuration of the octal latched peripheral driver).

Figure 4.5.1.2A - Pin configuration of the Octal Peripheral Driver (DP8311N)



Pin Names:

$\overline{\text{CLR}}$ -» Clear. The latches are cleared (outputs off) with a logic 0 on this pin.

$\overline{\text{STR}}$ -» Strobe. At logic 0 the input dat latch is active.

DI0 - DI7 -» Data input from the data bus.

DO1 - D07 -» Data output to the CONTROL BOX.

Each bit of the control word (see section 5.1.2 for further details) produced by the CPU represents a specific target for a specific device in the control box (see figure 4.5.1.2A).

Figure 4.5.1.2A - Word control of the PCS

D7	»—» Peristaltic pump for alkali reagent.
D6	»—» Peristaltic pump for antifoam reagent.
D5	
D4	»—» Air pump.
D3	»—» Heater.
D2	
D1	
D0	»—» Peristaltic pump for acid reagent.

A specific device in the fermenter vessel can stays continuously ON if its correspondent bit in the control word is at logic 1 all the time. This procedure can be carried out using two different approaches.

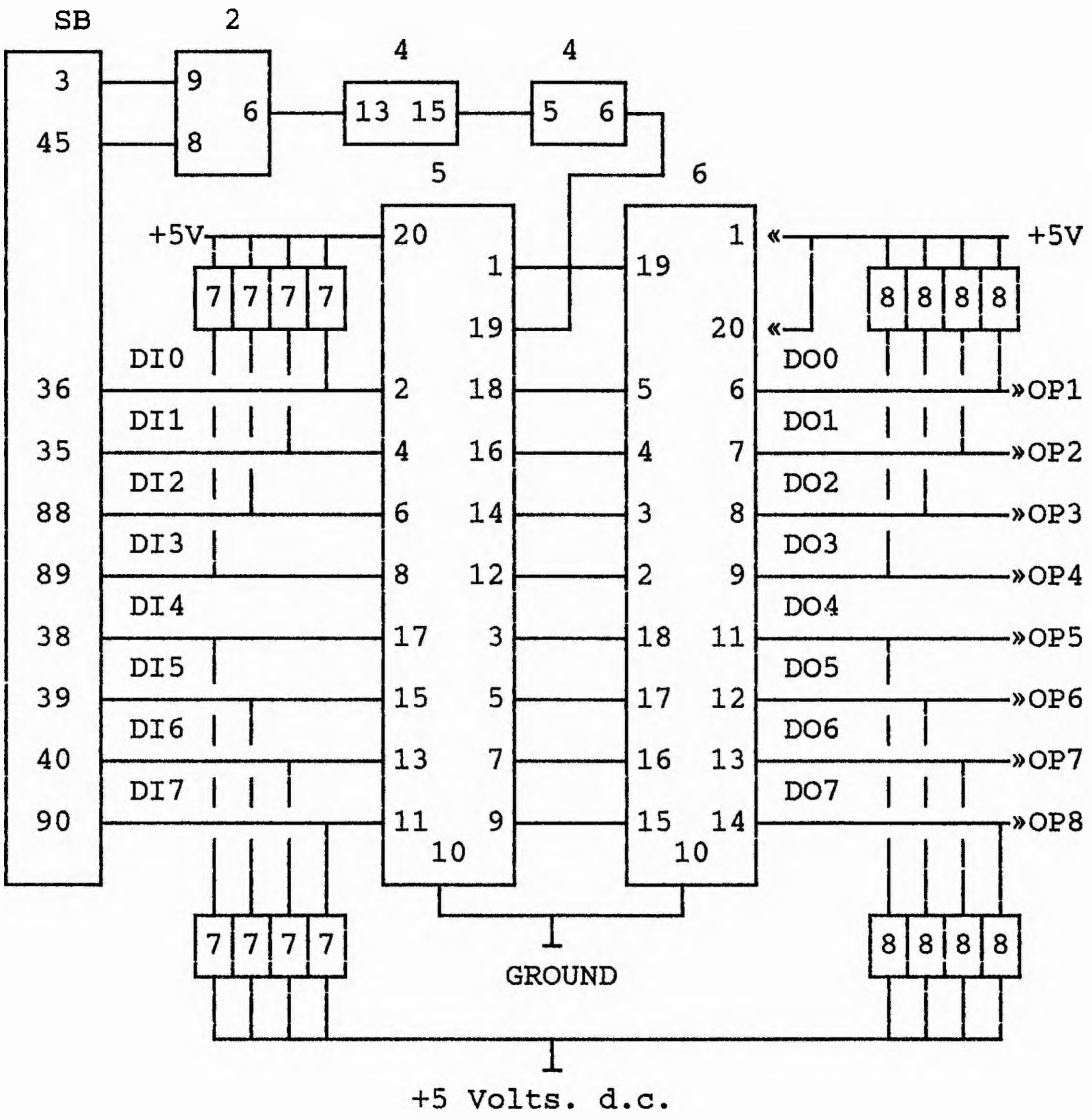
1 - Direct control from the CPU using a continue loop.

2 - The use of a loop system in a peripheral device, in this case the peripheral device is the octal latched peripheral driver.

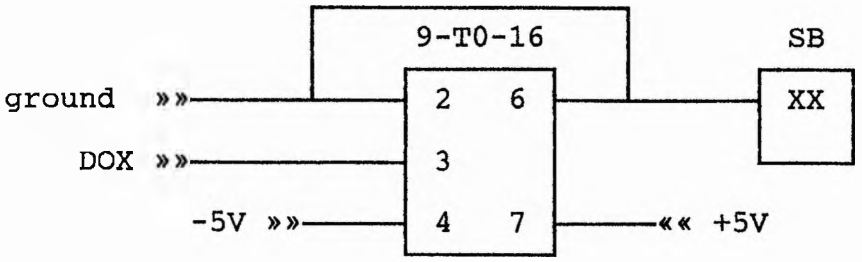
The first approach is impracticable, because it forces the CPU to stay in one loop all the time looking at only one parameter. So the second approach has been used, the octal latched peripheral driver can maintain a specific bit at logic 1 (in this case the device will be on) or at logic 0 (now the device will be off). (see software CONTROL for further details of the word control). Using this approach the CPU can control other parameters at the same time using the control word. The control word is generated by comparison between data from the fermenter sensors (see software CONTROL for further details) and set-point values from each

parameter. in figure 4.4.1.2b the circuit of the ON/OFF Switch system is described.

Figure 4.5.1.2b - ON/OFF Switch circuit.

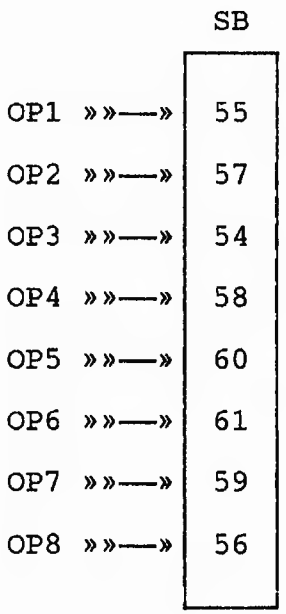


OPX -» represents a operational amplifier circuit for each output data (X varies from 1 to 8).



Where XX corresponds to a specific line in the bus system.

Each output line from the octal latch peripheral driver is fed to an operational amplifier circuit (see description above) prior to connection to the system bus (as showed below).



4.6 - Interface Boards (Second Block).

The second block of the Process Control System consists of four different interface boards as follows:

4.6.1 - pH interface board.

4.6.2 - Oxygen interface board.

4.6.3 - Temperature interface board.

4.6.4 - Biomass and Antifoam interface board.

Interfaces transform the usual low level and sometimes high impedance signal from sensors inside the fermenter vessel to a suitable level (0 to 2.55 V. d.c.) for analog-digital conversion.

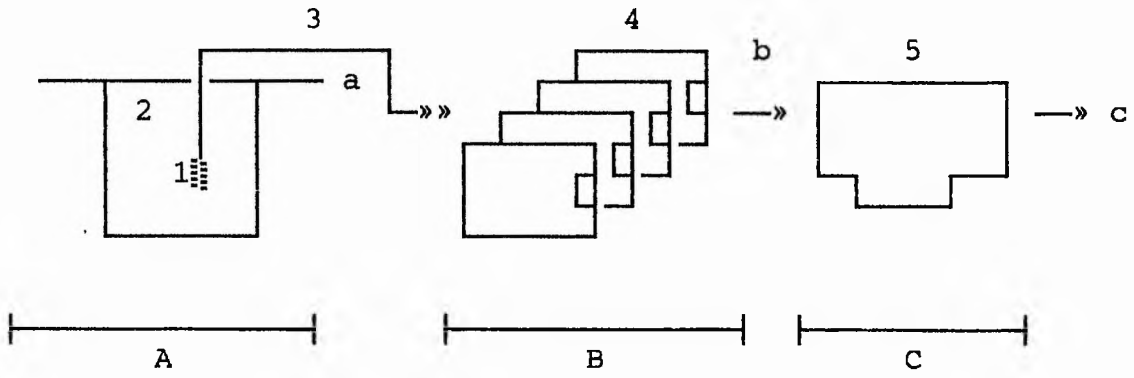
Figure 4.6 shows a general diagram of the interface system between the fermenter vessel and the ADS board. There are essentially three different parts, as follows:

A - The fermenter vessel;

B - The interface boards, which are connected by cables to the various sensors;

C - The ADS board which transforms analog signals came from the interface board (level 0 to 2.55 volts d.c.) to digital signals (0000 000 to 1111 1111), then puts the digital data in the data bus of the PCS.

Figure 4.6 - Diagram of Connections between the pH electrode and the PCS



Components: 1 - Electrodes

2 - Fermenter vessel

3 - Electrode cable (approx. 1 meter length)

4 - Interface boards

5 - ADS Board

Signals: a - Very low voltage or very low current from electrodes

b - 0 to 2.55 volts d.c.

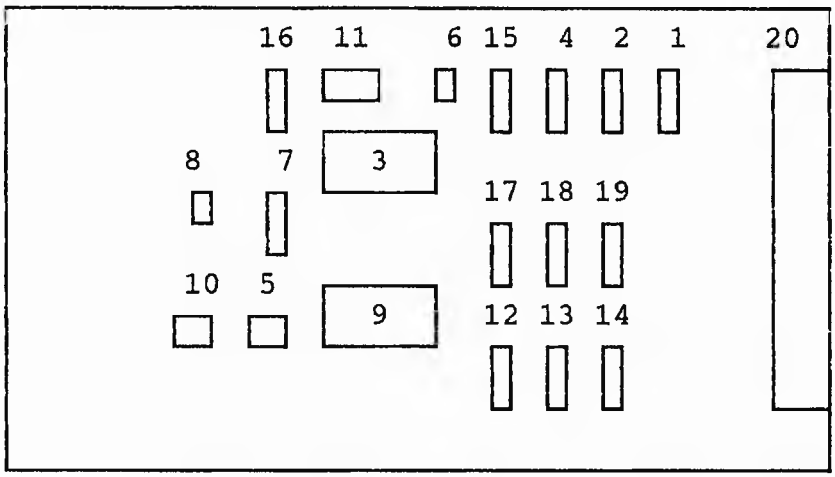
c - Digital signal to the PCS system (0000 0000 to 1111 1111.

4.6.1 - pH Interface board

4.6.1.1 - pH Interface circuit.

Figure 4.6.1.1 below represents a diagram of the pH interface board. It is a simple circuit mounted on a square pad board (RS 434-605). Pins and sockets were connected through wire using the wire-wrapping technique. This circuit converts the high-impedance signal from the ACWL 150 steam sterilisable pH electrode (Russell pH Limited, Auchtermuchty, Fife, Scotland), which is inside the fermenter vessel, to a suitable level to the Analog/Digital conversion.

Figure 4.6.1.1 - Diagram of the pH Interface board



Components:

1 - Polystyrene capacitor - 1000pF (RS)

3 and 9 - LF353 Wide Bandwidth DUAL JFET Input Operational Amplifiers (N).

2 and 4 - Resistors 470KΩ, 0.250W (RS)

13, 15 and 17 - Resistors 2.7KΩ, 0.250W (RS)

6 and 8 - Ceramic Capacitor - 10pF (RS)

7 and 16 - Resistor 47KΩ, 0.250W (RS)

10 - potentiometer 20-turn cermet trimmer 1KΩ (RS)

5 - Potentiometer 20-turn cermet trimmer 2KΩ. (RS)

11 - Potentiometer 25-turn cermet trimmer 100KΩ (RS)

12 and 14 - Resistor 10KΩ, 0.250W (RS)

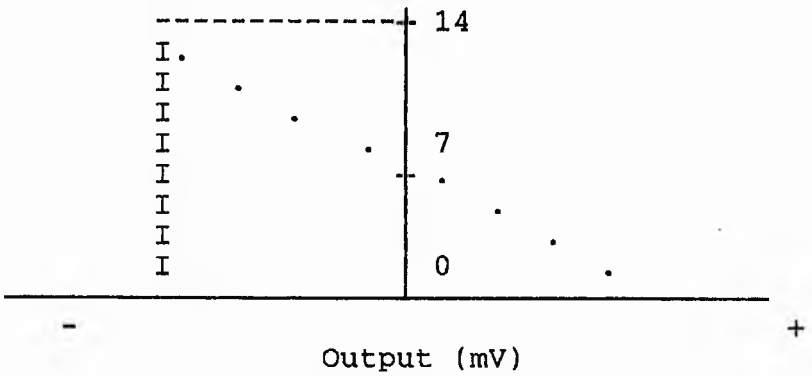
18 - Resistor 270Ω, 0.250W (RS)

19 - Zener diode 5.6V (RS)

20 - connector DIN 41612 - class III - 96 way. (RS)

The pH electrode signal varies with pH as below (figure 4.6.1):

Figure 4.6.1 - Electrode output signal



1 - When the pH is below 7, the signal is positive.

2 - When the pH is 7, the signal is zero.

3 - When the pH is above 7, the signal is negative.

This is important to understand how the pH interface works. Basically the interface consists of two parts.

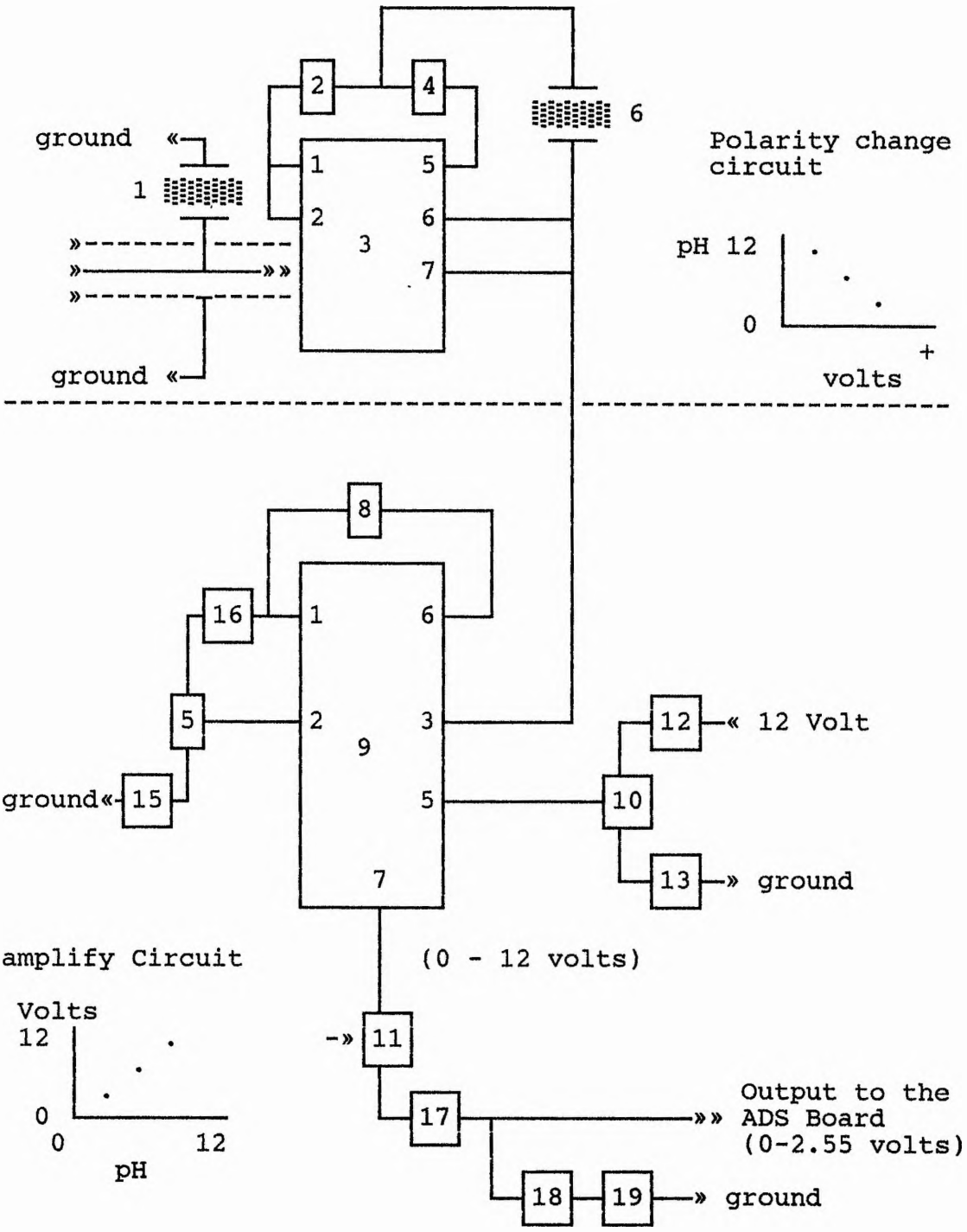
1 - A circuit to change negative voltages into positive voltages.

2 - An amplifier circuit.

Figure 4.6.1.1 shows the diagram of this interface, where the I.C. LF 353 (No. 3) is the heart of the circuit

which changes negative pH electrode signals to positive ones. These are then amplified by the second LF 353 (No.9) within the range 0 to 12 V corresponding from pH 0 to pH 12, then scaled to the range 0 to 2.55 volts.

Figure 4.6.1.1 - pH interface Circuit



4.6.1.2 - Calibration of the pH electrode.

The pH electrode is calibrated using three different buffers with pH values as following : pH 4.0, pH 7.0 and pH 9.0.

Procedure:

1 - A digital voltmeter is connected to pin 7 of the operational amplifier (No. 9 LF 353) which has an output voltage in the range 0 to 12 V, corresponding to pH 0 - 12.

2 - Insert the electrode in pH 7 buffer at 30°C, then adjust the potentiometer No.10 to obtain a output reading of 7.0V.

3 - Adjust readings of pH 7.0 in different temperatures (potentiometer No. 5) using data from another pH meter.

4 - The procedures described above will be carried out for other pH values (4 and 9).

4.6.2 - Oxygen Interface

The Oxygen interface amplifies the current from the sterilisable oxygen electrode type G-2 (Uniprobe, Cardiff, England) which produces a current output proportional to the oxygen partial pressure present in the fermenter medium. The electrode consists of two functional parts, the cathode and the anode. The oxygen diffuses across the teflon membrane into the electrolyte inside the electrode and reacts with the silver cathode and subsequently with the lead anode in the presence of the acetate electrolyte as follows:

$$\text{Cathode : } \tfrac{1}{2}O_2 + H_2O + 2e^- \longrightarrow 2OH^-$$

$$\text{Anode: } Pb \longrightarrow Pb^{2+} + 2e^-$$

$$\tfrac{1}{2}O_2 + Pb + H_2O \longrightarrow Pb(OH)_2$$

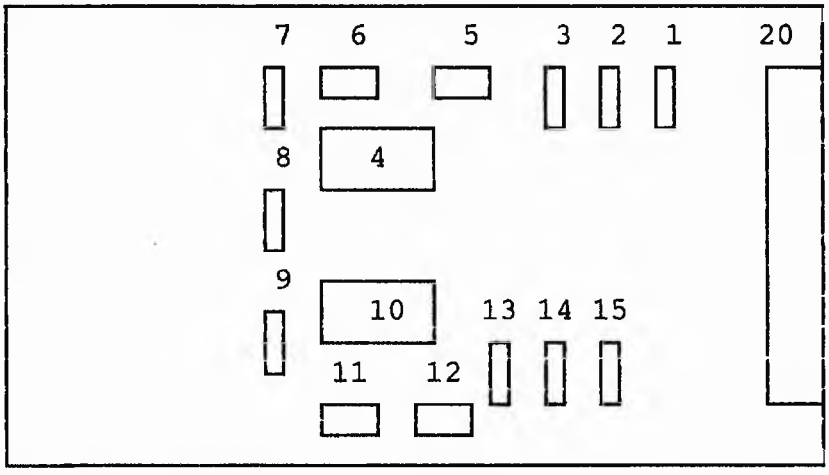
As described above, two components are consumed during the reaction, the lead and the acetate.

The current output of the electrode is proportional to the oxygen partial pressure, and not to concentration. Therefore diffusion of oxygen through the membrane is the rate determining factor and not the electrolytic capacity of the cell.

4.6.2.1 - Oxygen interface circuit.

The oxygen interface diagram described in figure 4.6.2.1 is a simple circuit. It consists of two circuits-the first amplifies the current from the electrode and then sends it to the second circuit, which arranges scaling in the range 0 - 2.55V for the ADS board. The scale which has been used in this interface is: 0% of dissolved oxygen corresponds to 0 V and 100% dissolved oxygen which represents 2.55 V.

Figure 4.6.2.1 - Diagram of the Oxygen Interface board.



Components :

1,2,7,8 - Resistors (10KΩ, 0.125W) (RS).

3 - Potentiometer (1KΩ) (RS).

4 and 10 - Operational amplifier 741CN (F).

5 and 11 - Potentiometer (10kΩ) (RS).

6 and 9 - Resistors (100KΩ, 0.125W) (RS).

12 - Potentiometer (100KΩ) (RS).

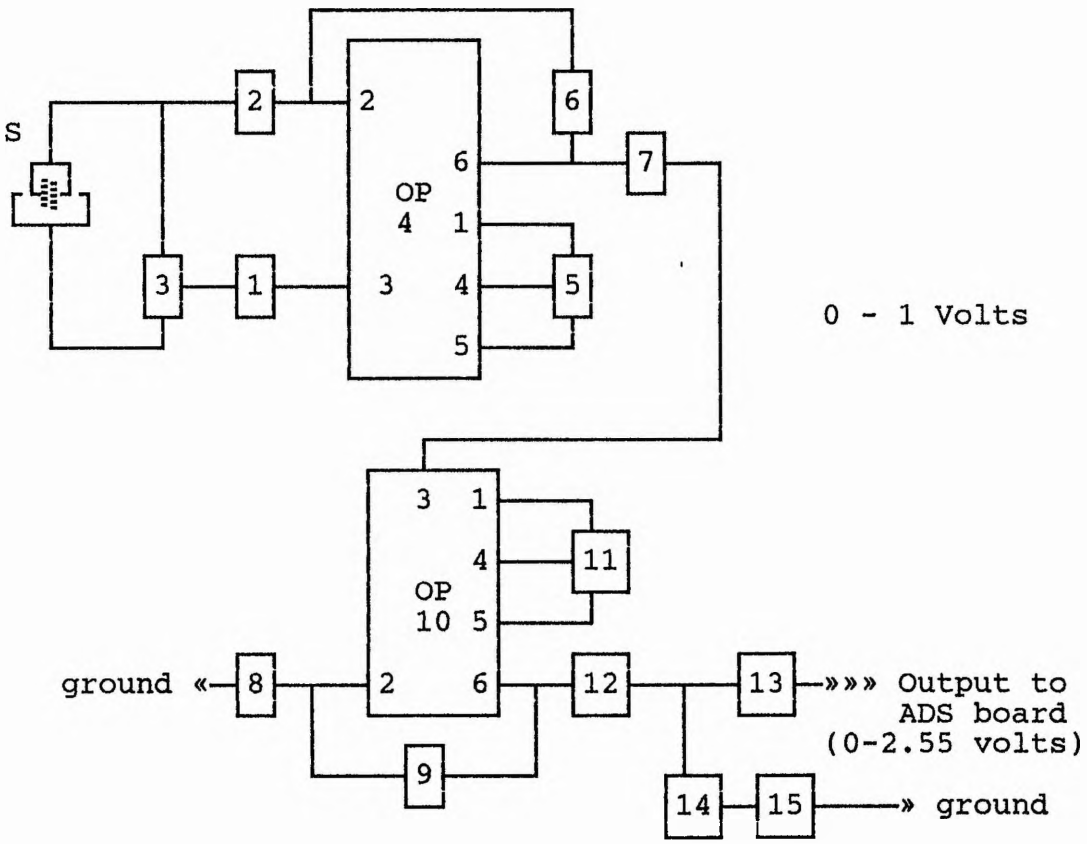
13 - Resistor (270Ω, 0.125W) (RS).

14 - Resistor (56Ω, 0.125W) (RS).

15 - Zener diode 5.6V (RS).

Figure 4.6.2.1a below shows the circuit of this interface which has two LM741C I.C. operational amplifiers. The first has been used to amplify the signal from the electrode from 0 to 1 V which corresponds 0 to 100 % dissolved oxygen saturation. The second has been used to convert the scale from 0 to 1 to 0 to 2.55 volts d.c..

Figure 4.6.2.1a - Oxygen Interface circuit



4.6.2.2 - Calibration of the Oxygen electrode.

The calibration of the oxygen electrode is carried out standardising the electrode at two fixed points, following the method described by Uniprobe Instruments (257). For convenience the 0% oxygen concentration and 100% saturation were used for calibration purposes.

A - Calibration at 0% of oxygen saturation.

A voltmeter is connected to pin 6 of the operational amplifier No.4, then the electrode is immersed in a oxygen free solution. In order to achieve this condition, two procedures have been carried out, as follows:

1. Procedure 1

   The electrode was immersed in a solution of 6% sodium sulphite (which chemically removes any oxygen present in the water) and the potentiometer No.5 adjusted (figure 4.6.2.1a) to read 0V in the voltmeter (this procedure should be carried out before the sterilisation).

2. Procedure 2

   The oxygen electrode is inserted in the fermenter vessel with 600 ml of medium and the vessel (with electrodes fitted) is sterilised at 15 psi (121°C, for 20 min). Immediately post-sterilisation, the oxygen concentration will be undetectable in the medium. The interface may then be calibrated to 0% oxygen saturation. This second procedure was generally used.

B - Calibration at 100% of oxygen saturation.

It is carried out using the same procedure described in the section above (calibration at 0% of oxygen saturation), but with the fermenter air pump on at maximum power and the stirrer speed at 1000 rpm. The air pump and agitation system were left on for 15 minutes and then the output voltage on pin 6 of the operational amplifier No. 4 was adjust to 10 V using potentiometer No. 3.

4.6.3 - Temperature Interface.

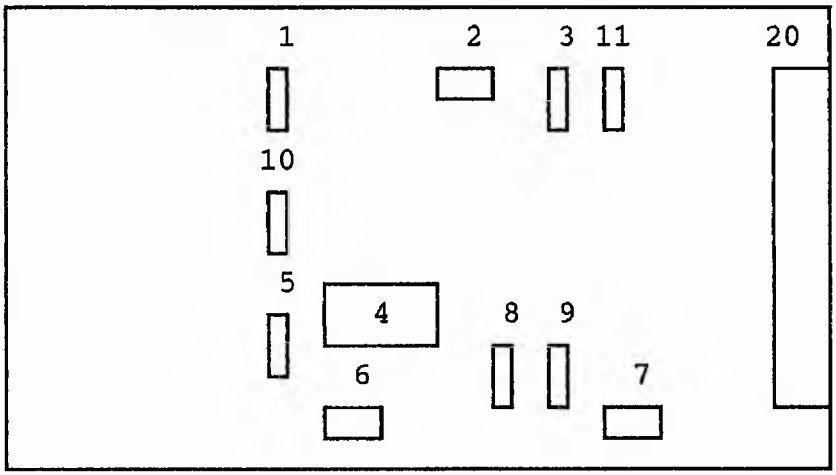
4.6.3.1 - Temperature Interface Circuit.

The temperature interface circuit which is described in figure 4.6.3A has been designed around the 590KH temperature transducer (RS Components, order number 308-309).

The 590KH is functionally a two-terminal I.C. temperature transducer which produces an output current proportional to absolute temperature. It operates in the range of -55°C to +150°C, with a nominal current output of 298.2 µA at + 25°C and a nominal temperature coefficient of 1µA/°C. It has a large range of operating voltage from 4V to 30V d.c.

In figure 4.6.3 the diagram of this interface in the PCS with the listing of components is described.

Figure 4.6.3 - Diagram of the Temperature Interface Board.



Components:

1 - Resistor (1KΩ, 0.125W) (RS)

2 - Potentiometer (5KΩ) (RS)

3 - Resistor (15KΩ, 0.125W) (RS)

4 - Operational amplifier 741CN (N)

5 and 11 - Resistors (10KΩ, 0.125W) (RS)

6 - Potentiometer (10KΩ, 0.125W) (RS)

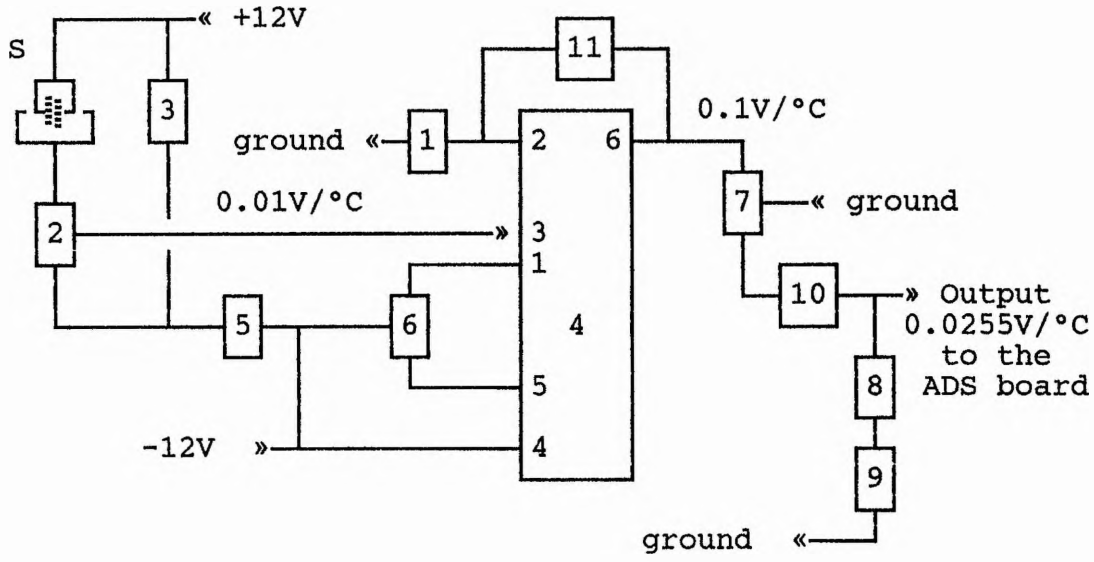
7 - Potentiometer (100KΩ, 0.125W) (RS)

8 - Resistor (56Ω, 0.125W) (RS)

9 - Zener diode 5.6V (RS)

10 - Resistor (270KΩ, 0.125W)

Figure 4.6.3A - Circuit of the Temperature Interface.



S -» Temperature sensor

4.6.3.1 - Calibration of Temperature electrode.

The temperature electrode circuit was calibrated using three different temperatures values as fixed points; 0°C, 30°C and 50°C. A voltmeter was connected in the pin 6 of I.C. No.4, then the sensor was immersed in a solution at 0°C and the potentiometer No. 2 (next to the sensor in figure 4.6.3A) was adjusted to give 0 V. The same procedure was carried out at 30°C (3 V) and at 50°C (5 V).

4.6.4 - Biomass Electrode System

The biomass electrode system (BES) consists of two discrete components; the electrode itself and the interface. The electrode consists of an emitter and a detector. The interface is made up of an amplifier circuit which amplifies the signal from the detector and transmits it to the ADS board, then to the CPU board where the processing of the data will be carried out.

4.6.4.1 - Electrode

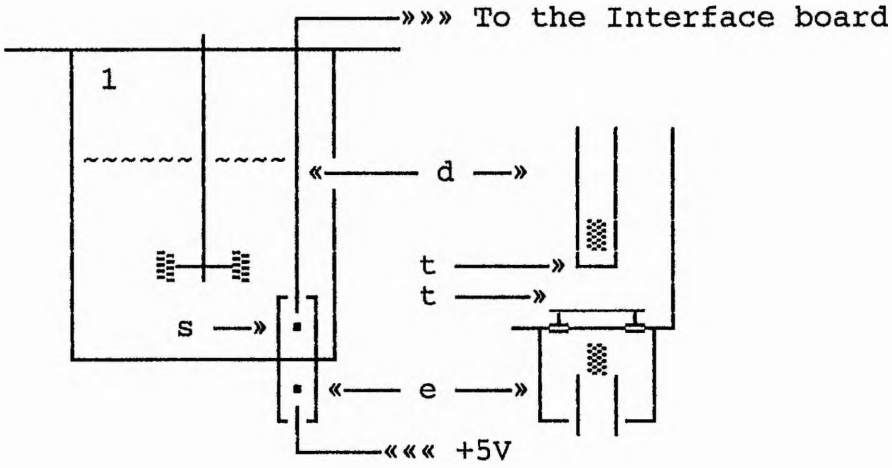
4.6.4.1.1 - Emitter

The emitter is a light emitting diode (LED) - the H500 (General Electric, U.S.A.). It is a GaAlAs diode which has a typically luminous intensity (TLI) of 500 mcd, and a spectral peak at 650 nm. It is fixed outside and to the bottom of the fermenter vessel, and its vertical position is aligned with the detector (Figure 4.6.4.1).

4.6.4.1.2 - Detector

The detector is the part of the electrode which is set up inside the fermenter vessel (Figure 4.6.4.1). It is an infrared detector (L14F1 - General Electric, U.S.A.), supersensitive NPN planar silicon photo-darligton amplifier. It is autoclavable and has a peak of light detection at 850 nm, but at 650 nm it retains 50% of its sensitivity, enough for the purpose of this system.

Figure 4.6.4.1 - Biomass electrode.



1 - Fermenter Vessel

s - Detector clamping device

e - Emitter

d - Detector

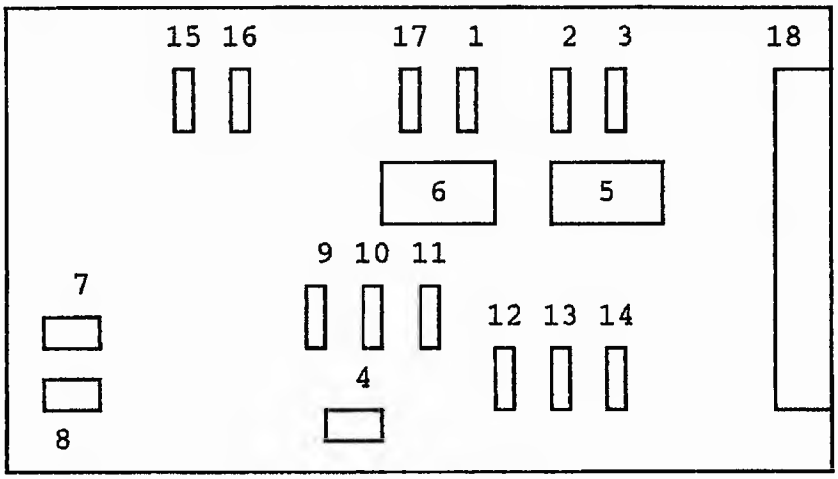
t - Teflon membrane.

4.6.4.2 - Interface

Figure 4.6.4.2 shows the diagram of the fourth interface board in the PCS. This board consists of two different circuits; a ~~the~~ biomass electrode interface which was described in the item before, and an antifoam circuit which will be described in item 4.6.5.

The interface board connects the biomass electrode to the microcomputer increasing the detector output (using a wide bandwidth JFET operational amplifier - LF351). The signal from the detector has a range from 0 to 0.55 V which is then scaled to the range 0 - 2.55V for processing by the ADS board (see figure 4.6.4.2a.).

Figure 4.6.4.2 - Diagram of the biomass and antifoam board.



Components:

1 - Resistor 100KΩ, 0.125W (RS)

2 and 3 - Resistors 1MΩ, 0.125W (RS)

4 - Potentiometer 10 ΩK (RS)

5 - Operational amplifier 741CN (N)

6 - Operational amplifier 351 (N)

7,8 and 11 - potentiometer 100KΩ (RS)

9 - Resistor 470KΩ, 0.125W (RS)

10 - Resistor 10KΩ, 0.125W (RS)

12 and 15 - Zener diodes 5.6V (RS)

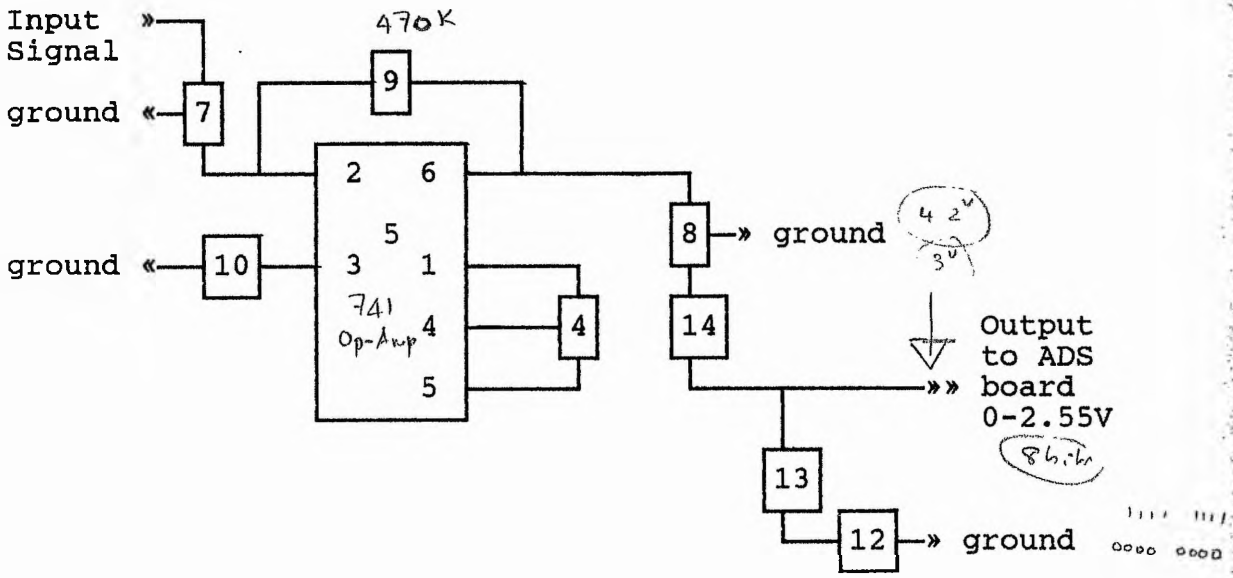
13 and 16 - Resistors 56Ω, 0.125W (RS)

14 and 17 - Resistors 270Ω, 0.125W (RS)

18 - Connecter DIN 41612 - class III - 96 ways (RS)

19 - Resistor 2MΩ, 0.125W (RS).

Figure 4.6.4.2a - Biomass electrode circuit



4.6.4.3 - Calibration of the Biomass electrode.

The Biomass electrode was calibrated as follows:

1 - The electrode was mounted in the fermenter vessel following three procedures:

1.1 - The emitter was fixed in the bottom of the fermenter using a nut fixed on the bottom of the fermenter vessel (see figure 4.1.4), then;

1.2 - The detector was clamped inside the fermenter vessel in vertical alignment with the exteriorly-mounted emitter and 0.5cm from the bottom of the glass vessel.

1.3 - With the PCS connected, the detector is adjusted by turning gently on the clamp until an output of 0.55V is registered (sterile medium in the vessel).

2 - Readings from the electrode during four growth batch growth experiments of the yeast Saccharomyces cerevisiae wild type Σ1278b (Temperature 30°C, Oxygen dissolved 30%, pH 5.0, in minimum medium as described in the section 3.2) were used to correlate biomass (as measured by the electrode) with the dry weight measurements.

3 - Using these data a simple equation (shown below) was developed. This equation represents the correlation between the readings from the electrode and spectrophotometer data (figure 4.6.4.3.).

Biomass=((AE + 0.001819) / (0.0862)) / 2.4

Where, AE represents the readings from the electrode, 2.4 is the factor to relate to dry weight.

3 - After this calibration, the system is suitable to operate with a biomass concentration up to 6g/l.

Figure 4.6.4.3 shows growth curve of Saccharomyces cerevisiae with data from the biomass electrode processed by the microcomputer and data from the spectrophotometerr readings.

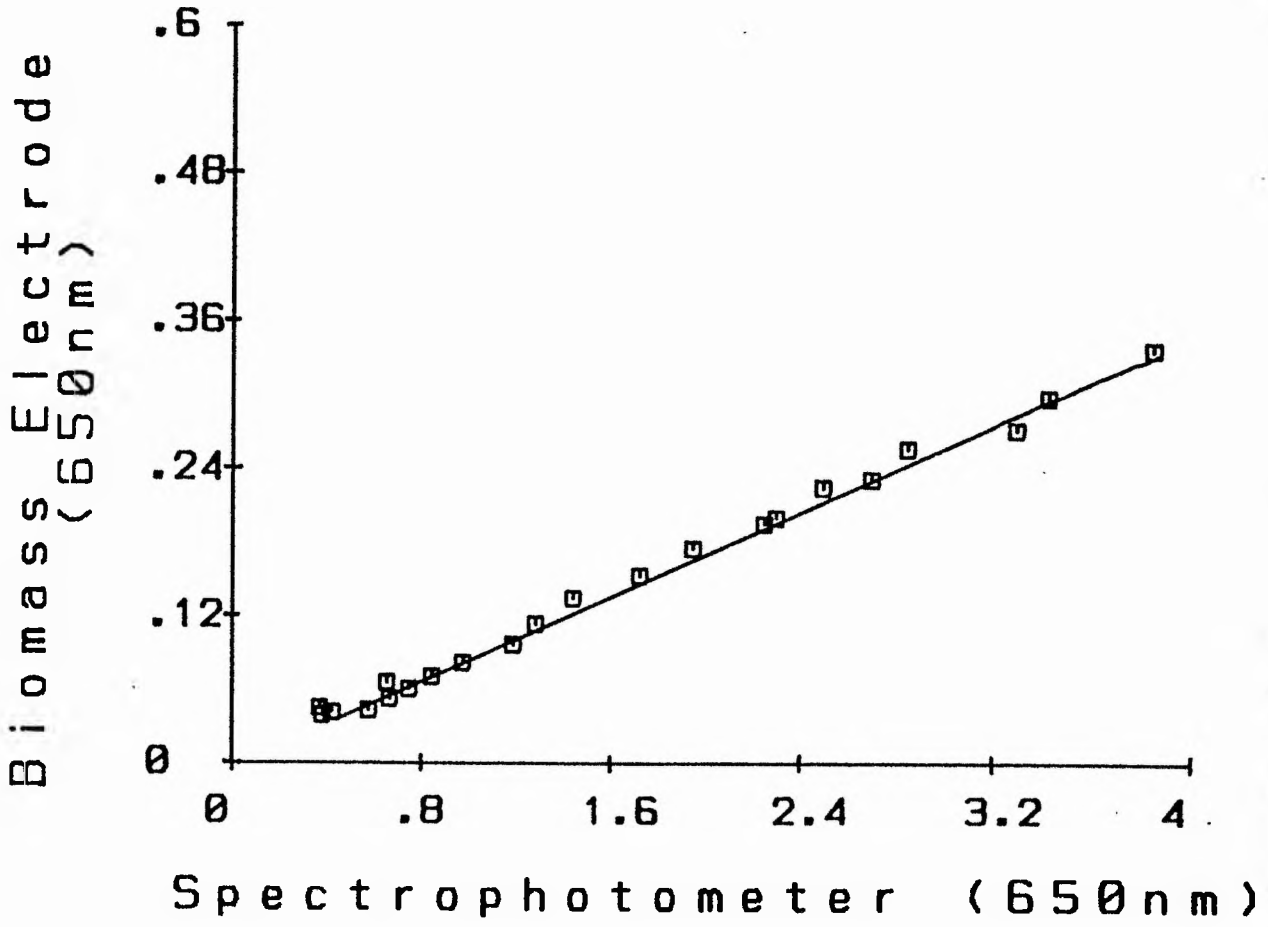


Figure 4.6.4.3 - Correlation between readings from the biomass electrode and readings from the spectrophotometer (readings being correlated with dry weight determinations of Saccharomyces cerevisiae wild type cells).

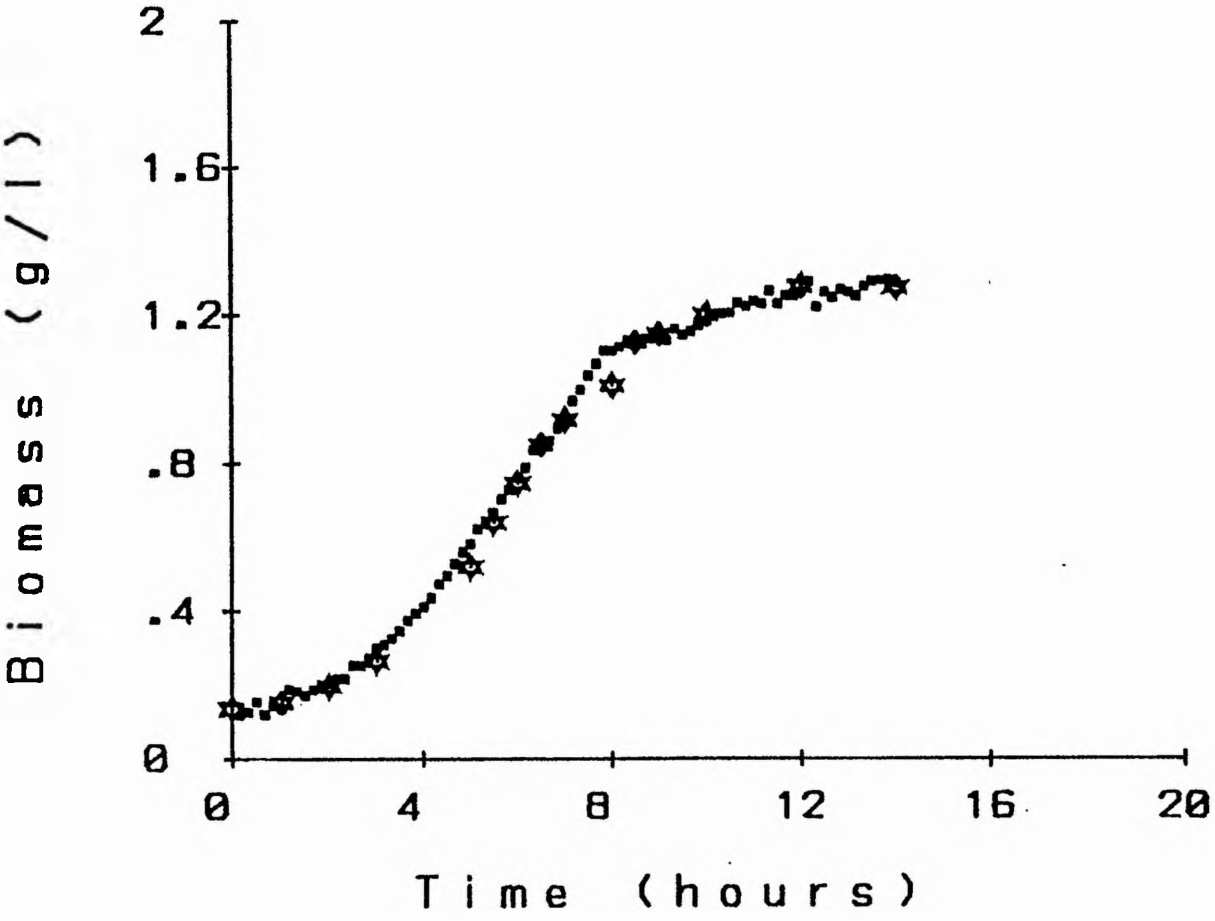


Figure 4.6.4.3a. - Correlation between spectrophotometer readings and readings from the biomass electrode (both at 650nm), of a Saccharomyces cerevisiae wild type cell culture.

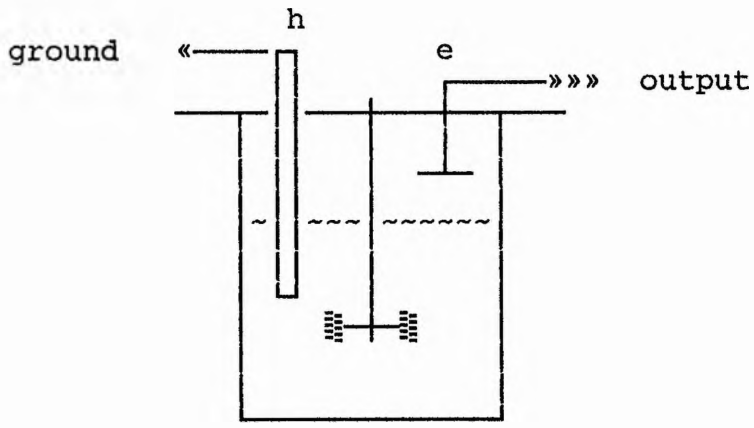
4.6.5. Antifoam Electrode System

The antifoam electrode system consists of an electrode in the fermenter vessel, and a interface board which amplifies the signal from this sensor before sending it to the ADS board.

4.6.5.1 - Antifoam Electrode

The electrode is a stainless steel tube which has a flat surface situated inside the fermenter vessel above the normal liquid surface (figure 4.6.5.1). Its flat end is used to detect the presence of foam through a small current which passes through the electrode when it is in contact with foam, using the heater as a ground (figure 4.6.5.1).

Figure 4.6.5.1 - Antifoam electrode

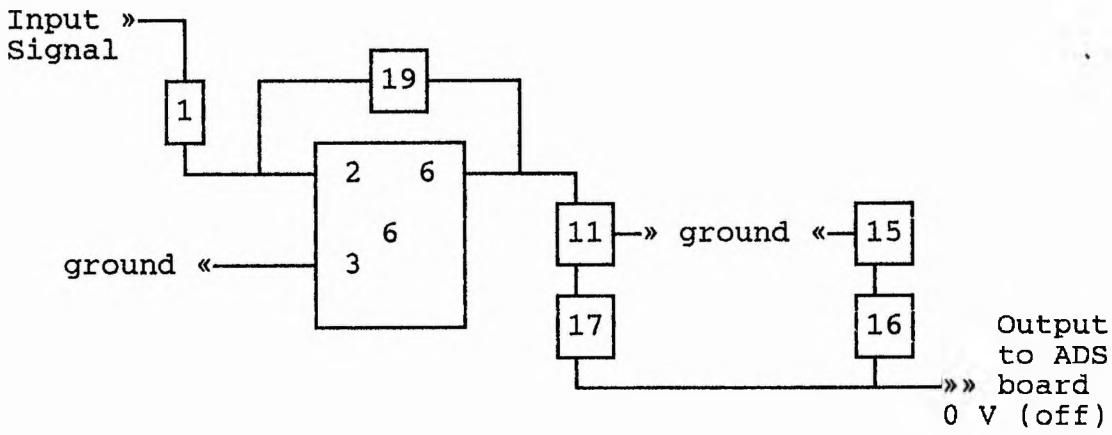


e -» Antifoam electrode

h -» heater

Figure 4.6.5.1a shows the antifoam electrode circuit, which consists of a simple inverting amplifier circuit using the 741 operational amplifier as the heart of the circuit.

Figure 4.6.5.1a - Antifoam electrode circuit.



4.6.5.2 - Calibration of the Antifoam electrode.

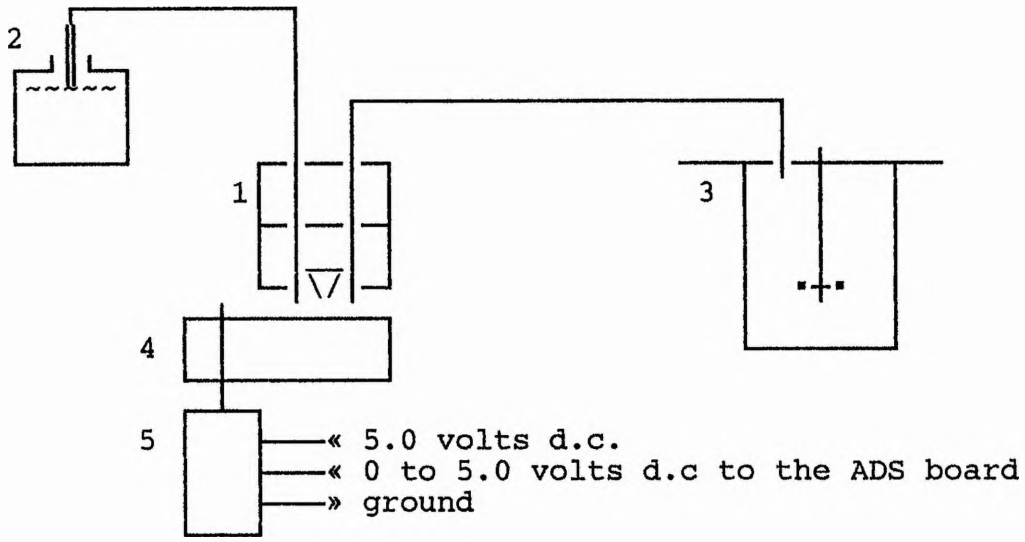
The antifoam system (electrode and the cirucit) is calibrated using the heater as ground (figure 4.6.5.1). When the electrode is in contact with foam, the voltage through the circuit decreases and consequently the voltage from the inverted amplify circuit rises. The signal from the circuit will rise above the set point fixed by the program in the PCS, and, consequently, the antifoam peristaltic pump will turned on pumping antifoam into the fermenter.

4.6.6 - Alkali Measurement Interface

During a growth curve experiment with the yeast Saccharomyces cerevisiae wild type Σ1278b or BC55 with or without plasmids, the value of the pH in the medium decreases. This occurs because of an increase in H^+ ions during the uptake of substrate by the cells. In continuous or batch culture experiments, the value of the pH of the medium was maintained at pH 5 by addition of alkali (KOH 2M) using the peristaltic pump P3 (see Box Control section 4.7).

The volume of alkali added to the fermenter during an experiment was measured by a simple system which records the duration of operation of the peristaltic pump by the turning a 10 turn digital counter connected with a precision potentiometer. Figure 4.6.6 shows the diagram of this system.

Figure 4.6.6 - Alkali measurement system



Components:

1 - Peristaltic pump P1

2 - Alkali reservoir

3 - Fermenter Vessel

4 - 10 Turn Digital Counter (F)

5 - Precision potentiometer 10ΩK,1W (B)

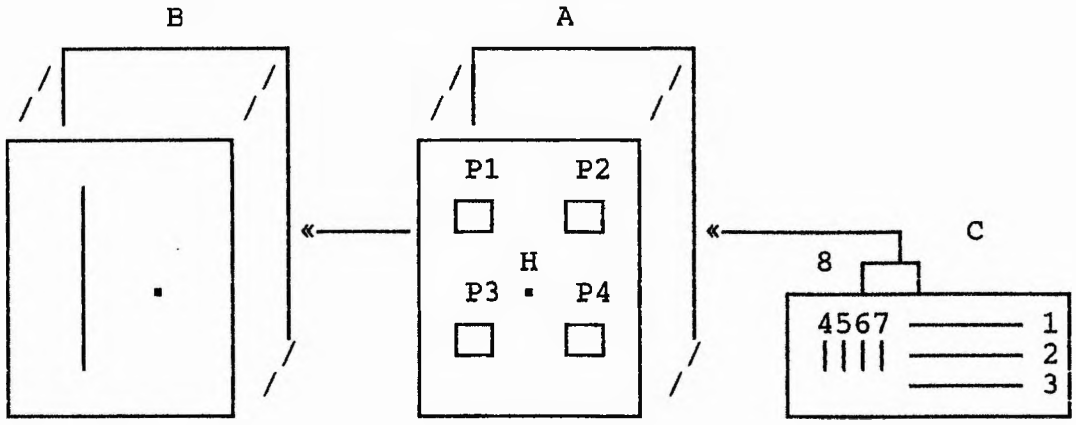
(B) - Beckman Instrument Ltd (Glenrothes, Fife, Scotland).

4.7 - Control Box

The control box is the part of the PCS which controls the environment conditions inside the fermenter vessel such as oxygen saturation, temperature, pH and foam using a series of on/off opto-triac switches (figure 4.7a) (MO 3020, Motorola, California, U.S.A.). These opto-triac switches can turn on a device which works from mains voltage using a digital signal.

Figure 4.7 shows the front view of the control box connected with PCS and the air pump. The control box has connected the heater and four peristaltic pumps: two for pH control, one for addition of antifoam and an extra pump.

Figure 4.7 - Front view of Control Box



Components: A - Control box

P1 - Peristaltic pump (Extra)

P2 - Peristaltic pump for acid

H - Heater connecter

P3 - Peristaltic pump for alkali

P4 - Peristaltic pump for antifoam

B - Air pump

C - Processor control system

1 - CPU board

2 - Memory board

3 - ADS Board

4 - Biomass and Antifoam interface

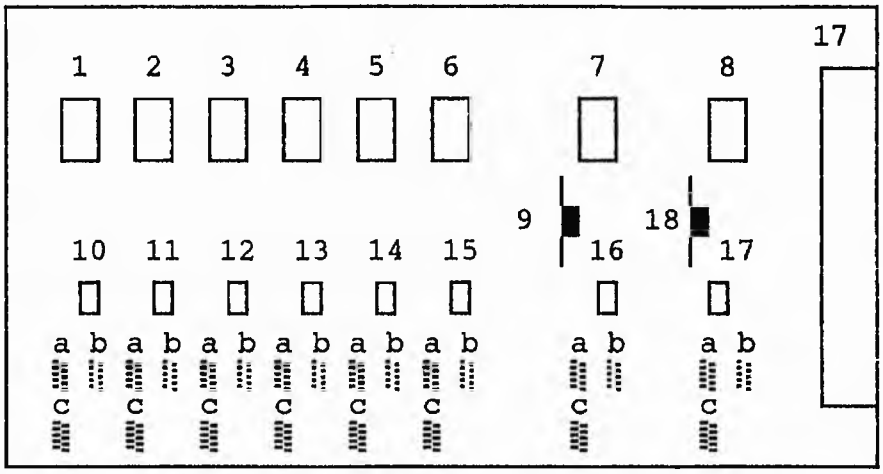
5 - Oxygen interface

6 - Temperature interface

7 - pH interface

8 - Connecter from ADS board to Box control.

Figure 4.7a - Diagram of the Control Box .



Components:

1 to 8 - Opto-coupled Triac 3020 (308-196) (RS)

9 and 18 - Triac SC 142D isolated, 8A 400V (F)

10 to 17 - Resistors 390Ω 1W (RS)

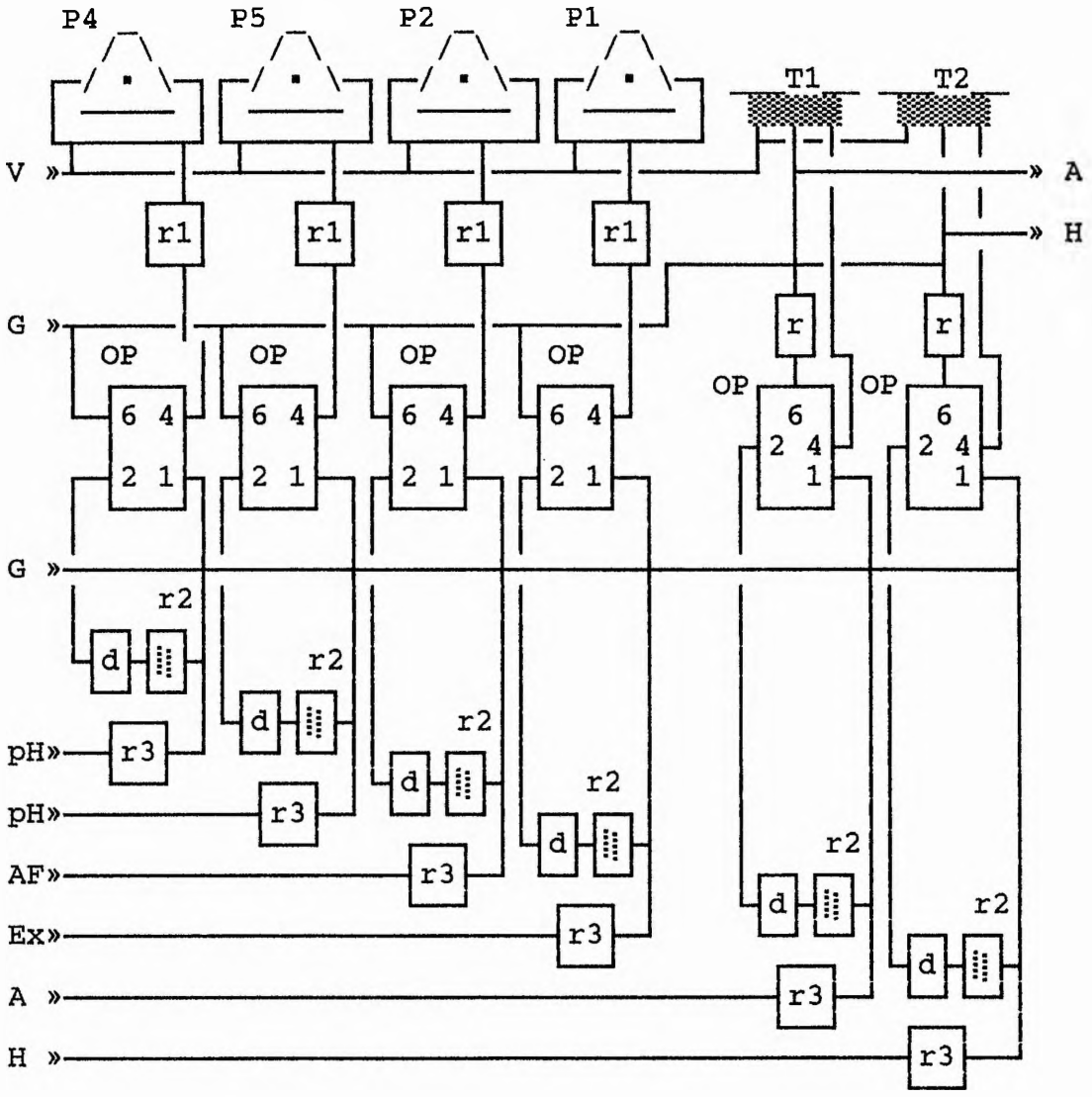
a - Zener diodes 3.6V (F)

b - Resistors 300Ω 0.5W (RS)

c - Resistors 270Ω 0.5W (RS)

Figure 4.7b describes the circuit of the control box. There are shown eight switch connections (opto-coupled triacs), however only six of them have been used in the PCS. The air pump and the heater circuits due to their high power consumption have been connected to the opto-triac through a triac SC 142D (which supports currents up to 8A).

Figure 4.7b - Circuit of the control box.



Components:

P1-» Extra peristaltic pump

P2-» Anti-foam peristaltic pump

P3-» pH peristaltic pump (alkali control)

P4-» pH peristaltic pump (acid control)

A -» Air pump connector

H -» Heater connector

V -» Mains power 220 V a.c.

G -» Ground

OP-» I.C. opto-coupled triac 3020

r1-» Resistors 390Ω, 1W

r2-» Resistors 300Ω, 0.250W

r3-» Resistors 270Ω, 0.250W

d -» Zener diode 3.6 V

T1 and T2 -» Triac SC 142D.

4.8 - PCS - BUS System

The PCS - Bus system consists of two different bus systems.

1 - Main boards bus system (MBBS);

2 - Interface boards bus system (IBBS).

4.8.1 - Main boards bus system

The MBBS utilizes a system similar to the S-100 bus, but with a few modifications concerning the connections of some lines (which will be discussed later). Physically, the S-100 bus is organised as a set of 100-contact edge connectors mounted on a common motherboard and wired in parallel.

The S-100 bus was originally designed to be used with the Intel 8080 microprocessor and it is very popular because: 1 - it was the first inexpensive microcomputer bus; 2 - It is used in hundreds of peripheral cards, including memories, terminal interfaces, graphics units, and speech synthesizers. However it is very poorly defined. Until recently there was no official definition concerning how the signals were intended to be used.

Because of its popularity, low price and the number of contacts, this bus system was chosen to be used in the PCS System.

Figure 4.8.1 shows the diagram of the MBBS which consists of three connecters (double-sided 50 way edge (RS - 468-939)) mounted on the back of the PCS frame and connections of the MBBS with two 25-way 'D' connecters which

are used to communicate between the PCS and the terminal or another computer through the two RS 232C links (see USART section for further details). These three connecters support the three main boards (CPU, Memory and ADS).

A complete listing of these bus signals provided by the CPU system board is shown below (figure 4.8). The "----" indicates that the particular line is not connected in the bus system.

Figure 4.8 - PCS - BUS sytem.

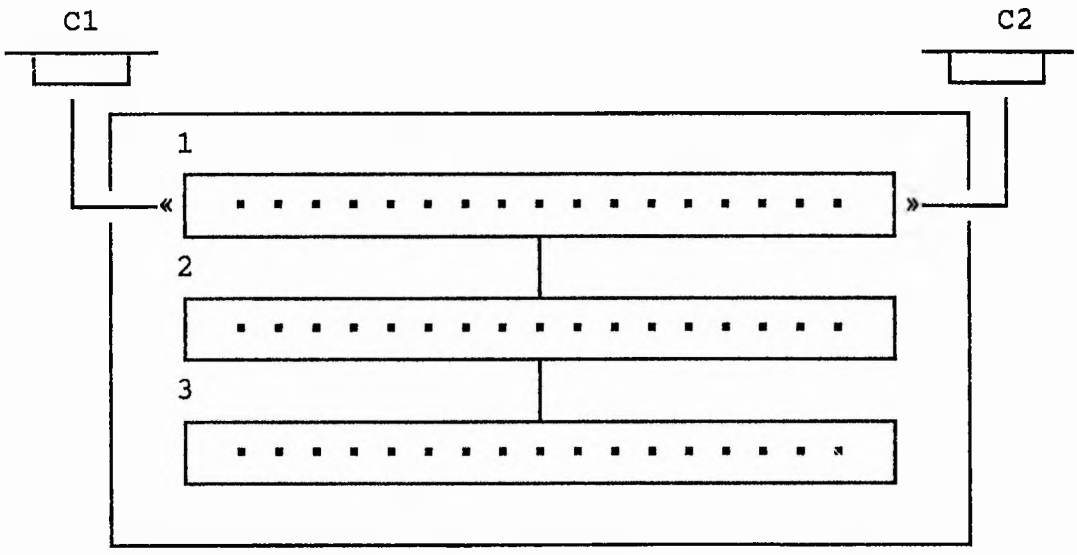
1. + 5 volts d.c	51. - 5 volts d.c.
2. -12 volts d.c	52. +12 volts d.c.
3. Switch System $\overline{CS}$	53. A/D $\overline{CS}$
4. ----	54. ----
5. ----	55. ----
6. ----	56. ----
7. ----	57. ----
8. ----	58. ----
9. ----	59. ----
10.----	60. ----
11.----	61. ----
12.----	62. ----
13.----	63. ----
14. G1RAM	64. ----
15. $\overline{MW}$	65. ----
16. G2RAM	66. ----
17. $\overline{MR}$	67. ----
18. ----	68. ----
19. ----	69. OUT USART1
20. G2ROM	70. IN USART1

21. ----
22. ----
23. ----
24. 500KHz
25. ----
26. ----
27. ----
28. ----
29. A5
30. A4
31. A3
32. A15
33. A12
34. A9
35. D1
36. D0
37. A10
38. D4
39. D5
40. D6
41. A/D 5
42. A/D 6
43. A/D 4
44. ----
45. $\overline{WR}$
46. $\overline{RD}$
47. ----
48. ----
49. ----
50. GROUND

71. OUT USART2
72. IN USART2
73. ----
74. RESET
75. ----
76. ----
77. ----
78. ----
79. A0
80. A1
81. A2
82. A6
83. A7
84. A8
85. A13
86. A14
87. A11
88. D2
89. D3
90. D7
91. A/D 3
92. A/D 1
93. A/D 0
95. A/D 2
96. ----
97. ----
98. ----
99. ----
100. GROUND

A/D No. -» Analog/digital device and No. indicates the number of one specific port in the Analog digital multiplexer.

Figure 4.8.1 - Diagram of the PCS-BUS system on the back of the PCS system



Components:

C1 - Connector 1

C2 - Connector 2

1 - CPU board

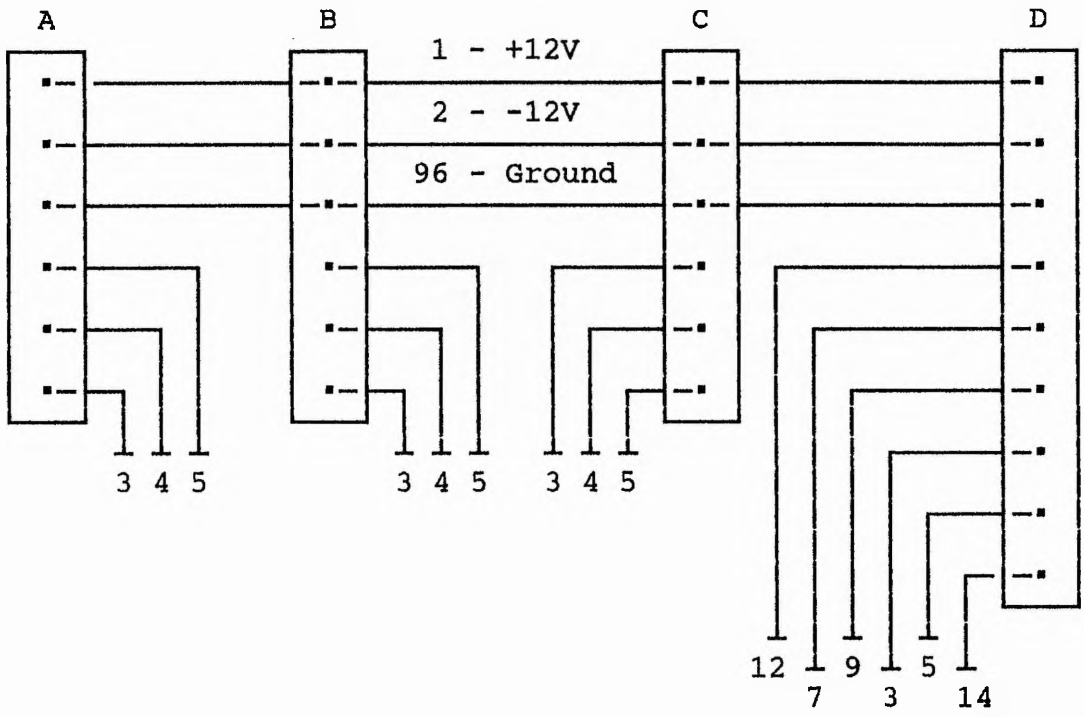
2 - Memory board

3 - ADS board

4.8.2 - Interface boards bus system

The IBBS utilises DIN 41612 (class III) connecters which have 96 signal contacts distributed in 3 rows of 32, but only a few contacts have been used as is showed in figure 4.8.2 below.

Figure 4.8.2 - Intefaces boards bus



Board: A - pH interface board

B - Oxygen interface board

C - Temperature interface board.

D - Biomass, Antifoam and Alkali measurement interface board.

Connections of A,B and C boards.

3 - Positive input signal from the electrode.

4 - Negative input signal from the electrode.

5 - Output signal to the ADS board.

Connection of the D board:

3 - Input signal from the Alkali Measurement device.

5 - Output signal from the Alkali measurement device circuit to the ADS board.

7 - Input signal from the biomass electrode.

9 - Output signal from the biomass circuit to the ADS board.

12 - Input signal from the Antifoam electrode.

14 - Output signal from the antifoam circuit to the ADS board.

4.9 - Power Supply.

A MOPS (Multi-Output Power Supplies) reference N100 (Farnell Electronic Components) was used as a power supply for this system.

5. Processor Control System - SOFTWARE.

The software used in the Processor Control System consists of three different packages as follows.

5.1 - Basic Input/Output System (BIOS) - It is eprom-resident providing input/output control for the PCS.

5.2 - Auxiliary Utilities - Disk resident on the second computer, these subroutines effect data conversion (hexadecimal (PCS) to decimal (second computer)).

5.3 - Data Processing - Disk resident on the second computer, these subroutines produce graphical output and carry out statistical analysis.

5.1 - BIOS (Basic Input/Output System)

The basic input/output system (BIOS) resides in an EPROM in the memory board providing control for the major I/O devices in the system.

The BIOS provides an operational interface to the system and relieves the user of the need to understand the characteristics of the hardware devices.

This section provides useful programming information

about the PCS-BIOS. A complete listing of the BIOS is given in Appendix A and Appendix B.

5.1.1 - Use of the BIOS.

The software consists of 3 different parts written in Z80 assembler codes (215, 216) as follows:

1. The PCS main system consists of subroutines which allow the system to alter, modify and load data from and to the RAM (see table 5.1.1).

2. The PCS control system is responsible for the control of conditions in the fermenter (e.g. oxygen level, pH, temperature and foam level). It also stores data relative to the rate of alkali addition, biomass concentration in RAM for further analysis (see table 5.1.1).

3. The display system produces a Y-time graph on the vdu terminal connected to the PCS from data on a specific parameter (e.g. pH, oxygen, temperature, rate of alkali addition, biomass). The Y axis scales can be modified providing better sensitivity for data from a specific parameter (up to a 16-fold increase). The sensitivity control is determined by the value in the address DE35 (on a scale from 10H (= 1 fold) or 01H (=16 fold)).

Table 5.1.1 - Main subroutines of the PCS-BIOS

Address (Hex)	Name	BIOS subroutine
001B	Initilisation	INICIO
0074	Output	STOUT
00A3	Input	STIN
00FC	Display	CMDD
013A	Load	CMDC
016B	Executes	CMDE
0180	Alteration	CMDA
02B4	Real time clock	KMASTER
035C	Control	CONTROL
04E2	Graph	CONTD

5.1.2. - The Control Word.

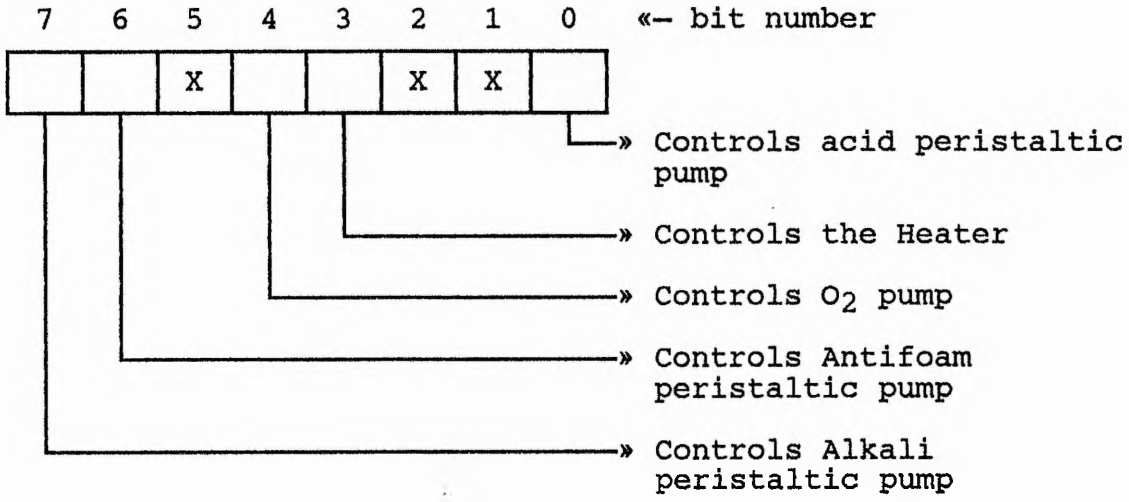
The Control Word (of bit length) controls the ON/OFF setting of the various devices (pumps, heater) which, in turn, control the fermenter parameters. This word, stored in address DFFC, may have its pattern altered according to parameter control requirements, by the CONTROL subroutine.

Diagram 5.1.2. shows which bits of the control word (bits 1,2 and 5 are unused) relate to which control devices. A bit of 1 indicates that a device is switched ON while a value of 0 means it is swithed OFF.

The control subroutine, which operates continuously in a loop, compares parameter data with established set points (see Appendix C) and alters the bit pattern in the control word accordingly. As an example, if oxygen saturation were to be found less than its set point with the O_2 pump off, bit 4 in the control word would be changed from 0 to 1. Each parameter value would be inspected in a full cycle of the

loop and the resultant overall control word restored in address DFFC at the end of the loop.

Diagram 5.1.2. - Significance of the bit value of the Control Word stored at DFFC.



Bit value Function:

1 --» Device ON

0 --» Device OFF

5.1.3 - Keyboard and video control

The BIOS provides a useful interface between the programmer and the hardware.

After the system has been reset, the 2 communication USARTs will be programmed by the BIOS, the USARTs being loaded with the mode operation command and the mode instruction, whose functions were described in the section 4.3.2.3.1. The BIOS is then able to communicate with the user and the hardware. A monitor prompt '>' now appears on the vdu screen (figure 5.1.3). A capital letter, representing a specific subroutine (table 5.1.3a), can now be entered followed by <CR>. If the letter entered is not present in the initialisation subroutine table (table 5.1.3a) a '?' will appear.

Table 5.1.3a - Composition of initialization subroutines

Code	Subroutine
K	KMASTER
V	TRSRRA
F	LOOPC
A	CMDA
C	CMDC
D	CMDD
E	CMDE
L	CMDL

Figure 5.1.3. - Prompt '>' on the vdu screen

The function of each subroutine in the initialisation set is explain below:

1 - Subroutine K.

Entering K on the keyboard executes the subroutine KMASTER (figure 5.1.3.a). This subroutine is responsible for the control loops and also for the real time clock (RTC) functions. It also resets all RTC registers as a procedure to synchronise them. The subroutine works using a loop technique. This technique consist of a sequence of commands running continuously between an initial and a final address. During one loop cycle, the KMASTER does several functions as shown below (figure 5.1.2).

Figure 5.1.3a - Initialisation of KMASTER subroutine.

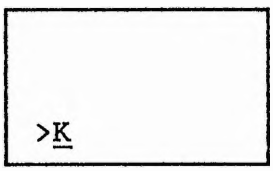
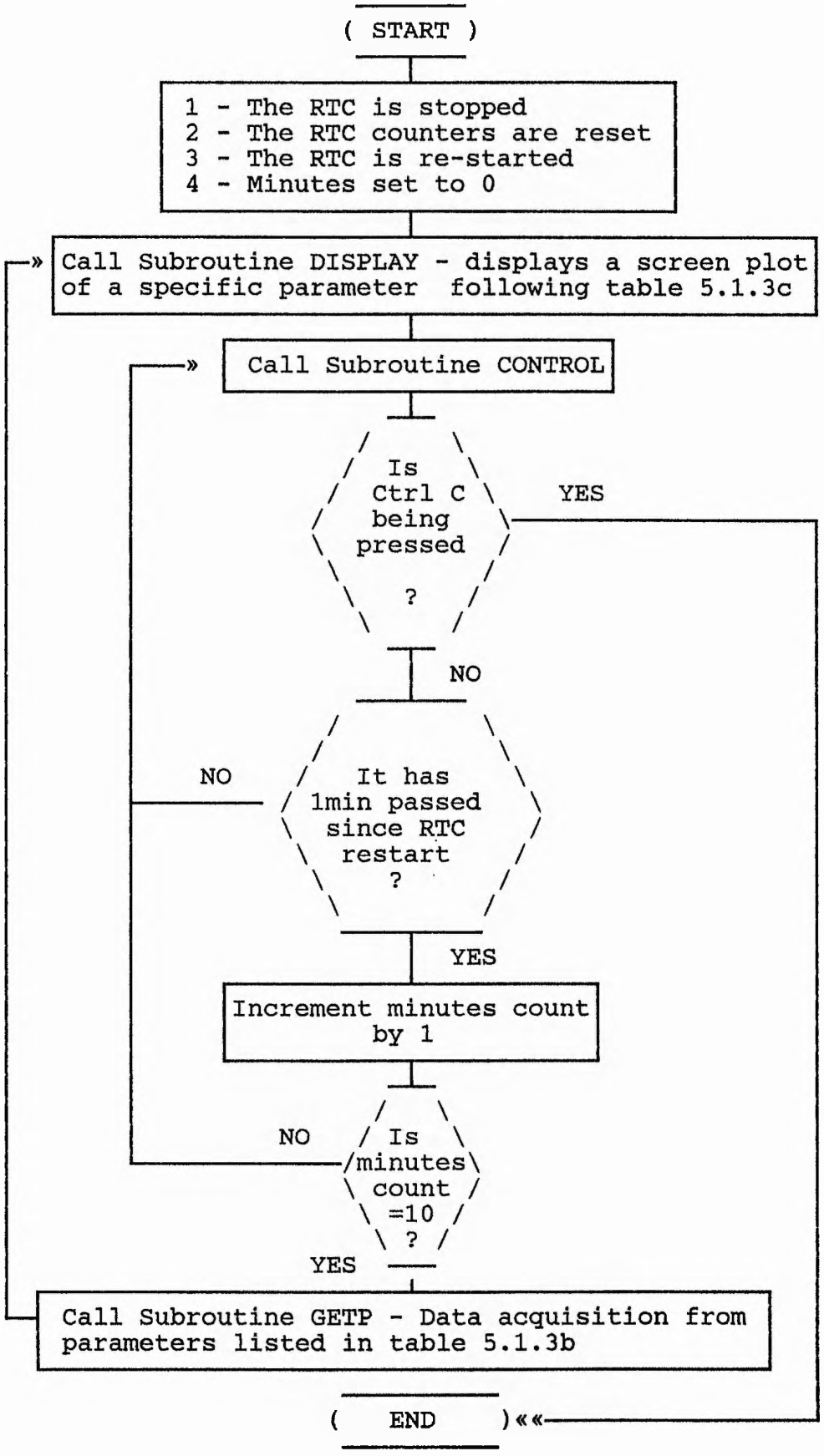


Figure 5.1.2 - Flowchart of KMASTER subroutine.



The KMASTER subroutine works all the time in a constant loop, execpt, after a CRTL C (^C), Reset, fault in the system or a power cut.

Note : KMASTER - Because it resets the Real Time Clocks and hence, in effect, turns the experiment and data collection, should only be run once at the begining of the experiment. It will be necessary to exit from KMASTER in order to transfer data from the PCS memory to the 2nd computer disc storage. For all subsequent control and data acquisition, subroutine F, which does not resets the Real Time Clocks, is invoked.

2 -Subrountine F.

Entering F against the monitor prompt will carries out essentially the same functions as KMASTER, except that F subroutine does not resets the Real Time Clock registers (figure 5.1.3b) and is used for control and acquisition subsequent to use of K subroutine as explained above. It should not be used otherwise to avoid random data in the RTC registers (which are initialised by K subroutine).

Figure 5.1.3b. - Initialization of F subroutine.

3 -Subroutine V.

This subroutine displays a screen plot of a specific parameter against time according to the code in table 5.1.3b.

To execute V:

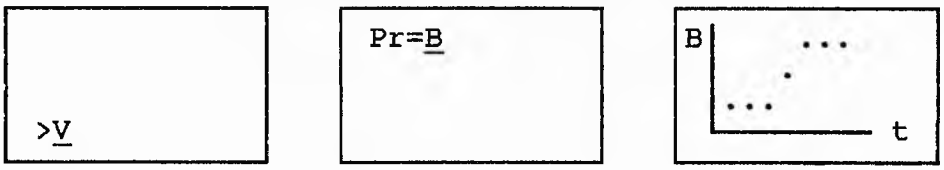
1. Enter Ctrl C (to exit from other subroutines)
2. Enter V
3. Enter with a code from the table 5.1.3b after the prompt 'Pr=' (see figure 5.1.3c).

Table 5.1.3b - Graph parameter codes

Parameter	code
Oxygen	O
Temperature	T
Biomass	B
Alkali system	M
pH	P

Figure 5.1.3c. - PCS graph subroutine V.

Example:Biomass plot



4 -Subroutine A

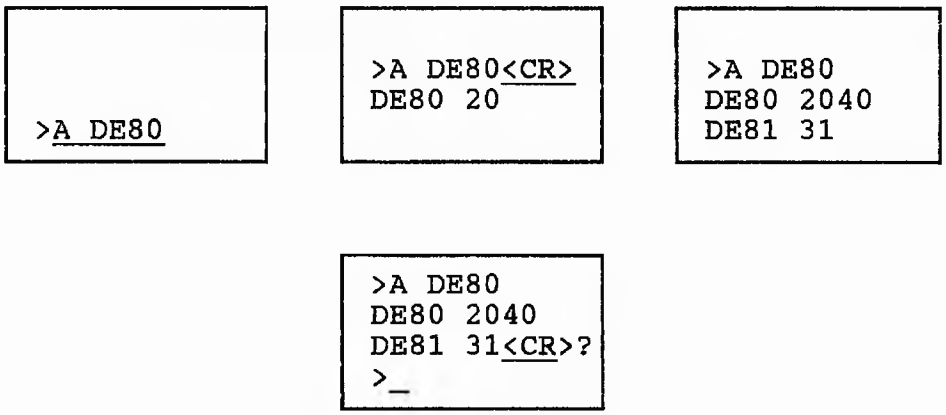
Replaces the content of a RAM memory address as described below:

To execute A:

1. Enter Ctrl C (to exit from other subroutines)
2. Enter A
3. Press space bar once.
4. Enter with address whose content is to be replaced
5. <CR>
6. Enter with the new memory contents

The subsequent address and its contents will be now displayed and may be similarly changed. To terminate and exit from A, press <CR>.

Figure 5.1.3d. - Execution of the subroutine A.



5 - Subroutine C

Loads information from the keyboard to the PCS RAM memory. It changes memory contents (like A) except that this subroutine allows the replacement of blocks of memory with 16 byte blocks being displayed per vdu line (figure 5.1.3e).

To execute C:

1. Enter Ctrl C
2. Enter C
3. Press space bar
4. Enter the initial address where the data is to be stored
5. <CR>
6. Enter data
7. <CR> to quit

Figure 5.1.3e - Subroutine C

```
>C DE80 <CR>
DE80 40 3A FF 4F 6B 52 78 89 90 CD D0 0B AF 3E EE 31
DE90 2A 4F<CR>?
>_
```

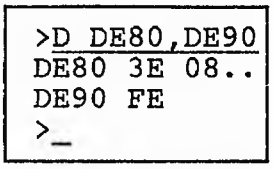
6 - Subroutine D.

This subroutine displays the contents of memory between 2 addresses (see figure 5.1.3f):

To execute:

1. Enter Ctrl C
2. Enter D
3. Press space bar
4. Enter with first address
5. Enter with final address
6. <CR>

Figure 5.1.3f. - Subroutine D. - Execution



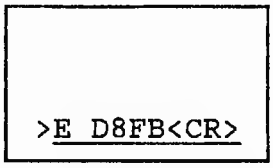
7 - Subroutine E.

This subroutine executes a sequence of instructions (program) starting from an initial address (figure 5.1.3g).

To execute:

1. Enter Ctrl C
2. Enter E
3. Press space bar
4. Enter with initial address
5. <CR>

Figure 5.1.3g. - Subroutine E. - Execution



5.1.4 - Data transfer.

Data transfer from the RAM memory of the PCS to another computer may be effected in 2 ways.

1. The USART1 port may be connected with another computer follown by execution of subroutine L (see section 5.1.4.2).

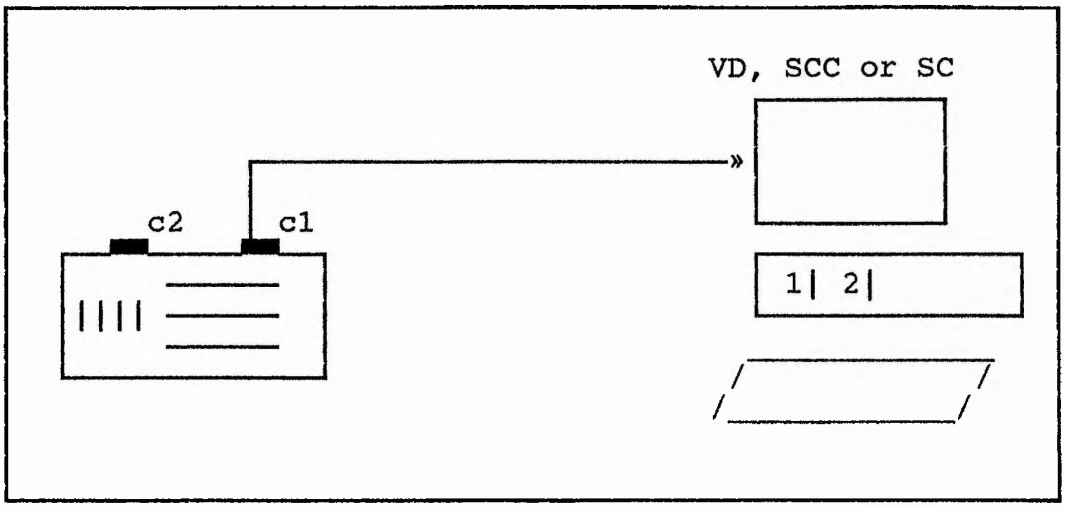
2. The USART2 port may be connected to another computer which is operating a communications package such as Kermit, PC-Talk III. The first procedure was carried out using as a second computer a Comart CP100 microcomputer which does not have any communication package. In the second procedure, an IBM-PC XT was utilised, using PC-TALK III software as a program link. (The description of how data transfer from the PCS RAM to the IBM disk drive will be explained in the section 5.2 - Auxiliary Utilities).

5.1.4.1 - Connection of the PCS with other computers.

The connection may be made in 3 different ways:

1 - Connecting directly the PCS with another microcomputer or a terminal through the connector on the top of the PCS frame on the right hand side using a RS 232C link (Figure 5.1.4.1a)

Figure 5.1.4.1a - Connection of the PCS with a terminal or microcomputer directly.



Parts: c1 - connector 1 of the PCS

c2 - connector 2 of the PCS

VD - Video terminal (Esprit II)

SC - Second computer (IBM PC XT)

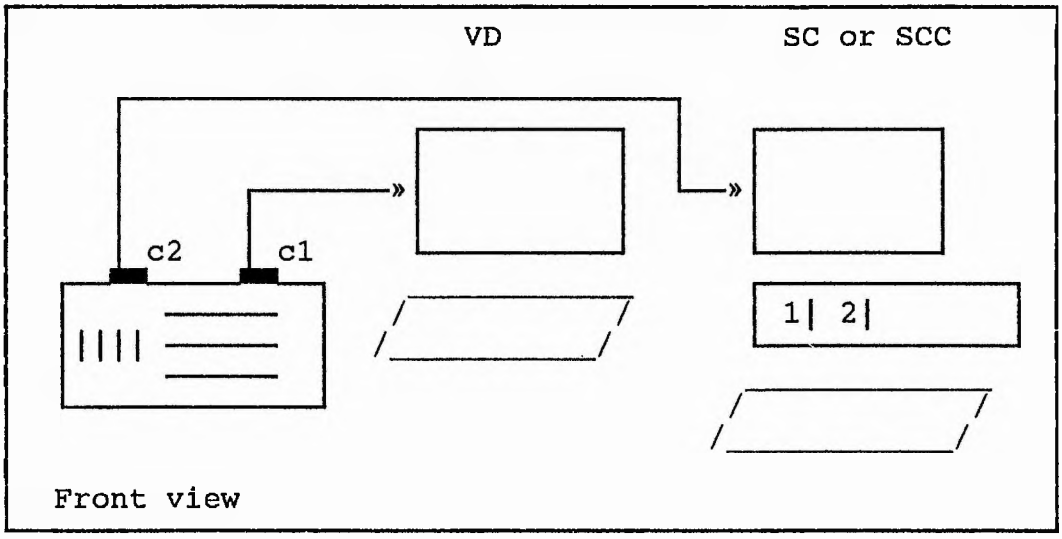
SCC - Second computer (Comart CP100)

1 - first disk drive

2 - second disk drive

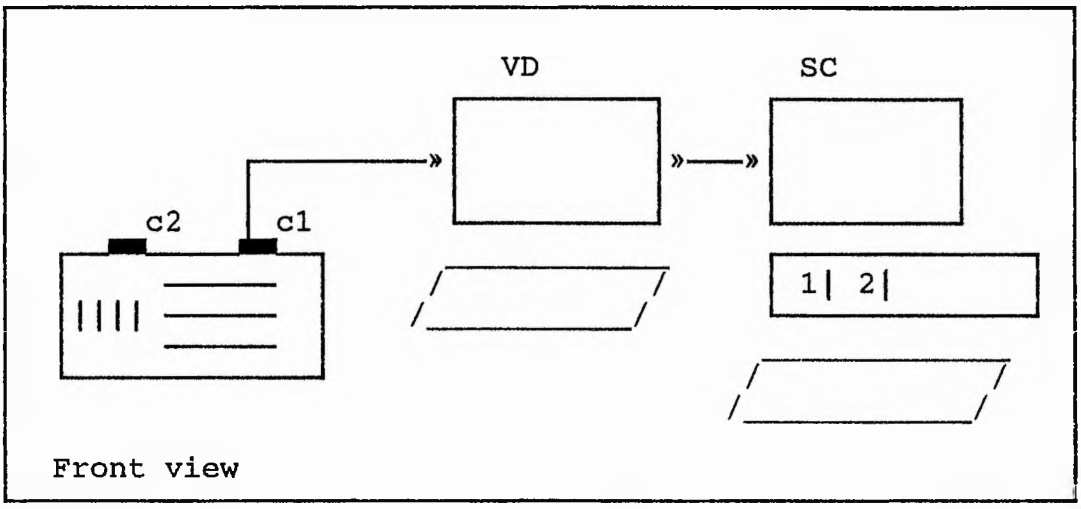
2 - Connecting the PCS with a terminal using the right hand side connector and another computer on the left hand side connector (figure 5.1.4.1b). In this case the data will be transferred using the subroutine L (see section 5.1.4.2).

Figure 5.1.4.1b - Connections between the PCS and the Second Computer using the subroutine L.



3 - Connecting another microcomputer to the PCS through a terminal using the right hand side connector as is shown in figure 5.1.4.1c below.

Figure 5.1.4.1c - Connections between the PCS and the Second Computer



5.1.4.2 - Second serial port communication package.

Communication between the PCS and another computer using this port is carried out by a combination of two programs, one in each computer. One will be run in the PCS (subroutine L) and program "STATUS-STOP" in the second computer (Figure 5.1.4.2.). The program used in the PCS is represented by the subroutine L described below. These programs are synchronised using a "status stop signal" from the second computer to the PCS.

Subroutine L . This subroutine is the subroutine D with an extra communication function. It can display data from a specific parameter (see table 5.1.4.2) at the same time in the PCS and in the second computer using the second USART (USART2). This routine is very useful if the computer which is used to process the data from the PCS does not have special communications software. In this case a BASIC program (STATUS-STOP - figure 5.1.4.2) will be run in the second computer in syncrony with the subroutine L in the PCS.

To execute:

1. Enter Ctrl C
2. RUN the subroutine STATUS STOP in the second computer.
2. Enter L
3. Against the "V=" prompt, enter a letter code from the table 5.1.4.2. corresponding to the selected parameter whose data is to be transferred.

4. After transmissiom press the reset button of the PCS

Table 5.1.4.2 - Parameters of subroutine L

Code	Parameters
O	Oxygen
T	Temperature
P	pH
B	Biomass
M	Alkali system

Figure 5.1.4.2a. - Execution of L subroutine

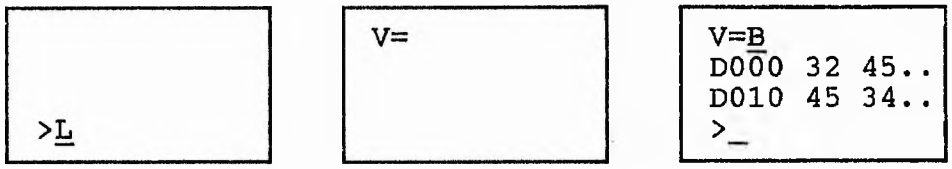


Figure 5.1.4.2. - Listing of STATUS STOP subroutine.

```
10  INPUT "File";F$
50  OPEN "O",#1,F$
100 OUT &H2,&H2
200 X=INP (&H2)
300 IF X<128 THEN 500
400 X=X-128
500 PRINT CHR$(X);
550 PRINT #1,CHR$(X);
600 IF X=62 THEN 800
700 GOTO 100
800 END
```

5.1.4.3 - The First Serial Port Communication Package.

The software involved in this package consists of the subroutine D from the monitor system of the PCS, together with a communications package PC-Talk III running on a IBM - PC XT. In order to transfer a file from the PCS to an IBM microcomputer using these programs, the following sequence is used:

Procedure:

1. Connect the IBM-PC microcomputer to the PCS (directly or through a terminal - see item 5.1.4.1).
2. Load and run the PC talk III in the IBM.
3. Enter Alt D in the IBM-PC.
4. Set the communication parameters: Baund rate = 1200, word length = 7 bits.
5. The IBM-PC now works as both a terminal for the PCS and as another microcomputer.
6. Enter Ctrl C
7. Enter A
8. Press space bar
9. Enter the location address (from table 5.1.4.3) appropriate to the parameter in question. The contents of this memory location are the address of the last datum point saved. Make a note of this address.
10. Enter D
11. Enter the initial setup address of the parameter (see table 5.1.4.3)

12. Enter the address noted in 9 above
13. Enter Alt R
14. Enter the name of the file in which the data will be stored in the IBM-PC.
15. Enter <CR> (Data transfer is now taking place).
16. Enter Alt R (This closes the file)
17. After carrying out these procedures for each parameter restart the KMASTER subroutine.
18. Enter F
19. Enter <CR>

Table 5.1.4.3 - Location of the final addresses of the fermenter parameters.

Parameter	Location	Initial setup
pH	DE22	C000
Oxygen	DE24	C400
Temperature	DE26	C800
Alkali system	DE28	CC00
Biomass	DE2A	D000

The data are now stored on disk for further processing which will be carried out using a serie of different BASIC programs which make a hexadecimal -» decimal conversion and display the results in high resolution graphics on the IBM-PC with a dot matrix printout. Figure 5.1.4.3 shows a growth curve of the yeast Saccharomyces cerevisiae Σ1278b in batch culture, using the system described in this section.

5.2 - Auxiliary Utilities

5.2.1 - Hexadecimal to Decimal Conversion

Programs listed later are used to convert disc-stored hexadecimal parameter data to decimal data for further processing.

These programs are necessary because while PCS stores data in hexadecimal format the analysis programs which have been used in the second computer require decimal format.

This program (GRAPH1.BAS) which first converts hexadecimal data to decimal data should be run prior to any attempt at further analysis - especially if communication packages (Kermit - Columbia Center, Columbia, U.S.A. or PC-TALK - The Headlands, California, U.S.A.) have been used for data transfer to an IBM-PC (or close clone).

Program GRAPH1.BAS

```
100 CLS
200 SCREEN 0
300 COLOR 2,0,6
400 PRINT "Parameter"
500 PRINT
600 PRINT "   Oxygen  - O"
700 PRINT "   pH      - P"
800 PRINT "   Biomass - B"
900 PRINT "   Ammonia - A"
1000 PRINT
1100 PRINT " Enter with a capital letter, please! ": INPUT L$
1200 IF L$<>"O" THEN GOTO 1500
1300 L1$="OXYSUP.BAS"
1400 GOTO 2400
1500 IF L$<>"P" THEN GOTO 1800
1600 L1$="PHSUP.BAS"
1700 GOTO 2400
1800 IF L$<>"B" THEN GOTO 2100
1900 L1$="BIOSUP.BAS"
2000 GOTO 2400
```

```
2100 IF L$<>"A" THEN GOTO 2400
2200 L1$="AMMSUP.BAS"
2300 GOTO 2400
2400 REM
2500 IF L1$<>"O" THEN 100
2600 IF L1$<>"P" THEN 100
2700 IF L1$<>"B" THEN 100
2800 IF L1$<>"A" THEN 100
2900 PRINT "Experiment =";:INPUT R$
3000 OPEN "I",#1,R$
3100 DIM Y(300)
3200 INPUT #1,R1$
3300 K=K+1
3400 ON ERROR GOTO 3800
3500 INPUT  #1,T$(K)
3600 Y(K)=LEN(T$(K))
3700 GOTO 3300
3800 CLOSE
3900 OPEN "O",#1,"WORK.DAT"
4000 DIM X(K*255),X1(K*255)
4100 PRINT #1,R$
4200 K1=1
4300 FOR J=1 TO K
4400 K1=4
4500 FOR I=K1 TO Y(J)-2
4600 X1(I)=ASC(MID$(T$(J),I,1))
4700 IF X1(I)<>32 THEN GOTO 5300
4800 J1=J1+1
4900 X(I)=ASC(MID$(T$(J),I+1,1))
5000 X(I+1)=ASC(MID$(T$(J),I+2,1))
5100 PRINT #1,X(I);X(I+1);
5200 ON ERROR GOTO 5400
5300 NEXT I,J
5400 REM
5500 CLOSE
5600 CHAIN L1$
5700 END
```

5.2.1.1 - Subroutines for use with GRAPH1.BAS

In this section support programs (XXXSUP.BAS - see table 5.2.1.1) for the five parameters are listed. Data of four parameters (Oxygen, pH, Alkali added and biomass) are stored on disk. The temperature file was used only as a on the system temperature control.

These programs produce screen plot (Y-time) (see figure 5.2.1.1). If want a printout on a dor matrix printer, enter Shift PrtSc.

Table 5.2.1.1 - Programs XXXSUP.BAS.

Parameter	Program
Biomass	BIOSUP.BAS
Oxygen	OXYSUP.BAS
pH	PHSUP.BAS
Alkali	AMMSUP.BAS

AMMSUP.BAS

```
100 DIM R3(1500),R4(1500),X(1500),Y(1500)
200 OPEN "I",#1,"WORK.DAT"
300 INPUT #1,B$
400 KEY OFF
500 REM
600 T=T+1
700 INPUT #1,X
800 ON ERROR GOTO 3700
900 IF X>64 THEN 1200
1000 X=(X-48)*16
1100 GOTO 1300
1200 X=(X-55)*16
1300 R=X
1400 INPUT #1,X
1500 IF X>64 THEN 1800
1600 X=(X-48)
1700 GOTO 1900
1800 X=(X-55)
1900 R1=X
2000 R3(T)=R+R1
2100 INPUT #1,X
2200 IF X>64 THEN 2500
2300 X=(X-48)*16
2400 GOTO 2600
2500 X=(X-55)*16
2600 R5=X
2700 INPUT #1,X
2800 IF X>64 THEN 3100
2900 X=(X-48)
3000 GOTO 3200
3100 X=(X-55)
3200 R6=X
3300 R7=R5+R6
3400 R8=((R7*10)/100))/10
3500 R4(T)=((R3(T)+R8)*11.2)/255
3600 GOTO 500
3700 CLOSE
3800 OPEN "O",#1,"AMM.DAT"
3900 FOR I=1 TO T
4000 IF R4(I)=0 THEN 4700
4100 IF R4(I)-R4(I-1)<2 THEN 4300
```

```
4200 J=Y(I-1)
4300 Y(I)=ABS(11.2-R4(I))+J
4400 X(I)=R4(I): X(I)=Y(I)
4500 PRINT I/6,X(I)
4600 PRINT #1,I/6,(X(I))
4700 NEXT I
4800 CLOSE
4900 STOP
5000 CLS
5100 SCREEN 2
5200 LINE (80,24)-(80,150),1
5300 LINE (80,150)-(500,150),1
5400 FOR K=1 TO 130 STEP 21
5500 LINE (75,23+K)-(80,23+K),1
5600 NEXT K
5700 FOR K2=1 TO 450 STEP 30
5800 LINE (79+K2,156)-(79+K2,151),1
5900 NEXT K2
6000 ON ERROR GOTO 6100
6100 FOR J=1 TO I
6200 X(J)=(X(J)*126)/120
6300 PSET (80+J,150-X(J)),1
6400 NEXT J
6500 ON ERROR GOTO 6600
6600 REM
6700 LOCATE 2,7:PRINT "alk (ml)"
6800 LOCATE 3,7:PRINT 6
6900 LOCATE 19,7:PRINT 0
7000 LOCATE 21,10:PRINT 0
7100 LOCATE 21,17:PRINT 10
7200 LOCATE 21,24:PRINT 20
7300 LOCATE 21,32:PRINT 30
7400 LOCATE 21,39:PRINT 40
7500 LOCATE 21,47:PRINT 50
7600 LOCATE 21,54:PRINT 60
7700 LOCATE 21,62:PRINT 70
7800 LOCATE 22,32:PRINT "time (hours)"
7900 LOCATE 23,7:PRINT "Exp = ";B$
8000 INPUT D
8100 CHAIN  "ENTER.BAS"
8200 END
```

OXYGEN.BAS

```
100 DIM R3(1500),R4(1500),X(1500),Y(1500)
200 OPEN "I",#1,"WORK.DAT"
300 INPUT #1,B$
400 KEY OFF
500 REM
600 T=T+1
700 INPUT #1,X
800 ON ERROR GOTO 3700
900 IF X>64 THEN 1200
1000 X=(X-48)*16
```

```
1100 GOTO 1300
1200 X=(X-55)*16
1300 R=X
1400 INPUT #1,X
1500 IF X>64 THEN 1800
1600 X=(X-48)
1700 GOTO 1900
1800 X=(X-55)
1900 R1=X
2000 R3(T)=R+R1
2100 INPUT #1,X
2200 IF X>64 THEN 2500
2300 X=(X-48)*16
2400 GOTO 2600
2500 X=(X-55)*16
2600 R5=X
2700 INPUT #1,X
2800 IF X>64 THEN 3100
2900 X=(X-48)
3000 GOTO 3200
3100 X=(X-55)
3200 R6=X
3300 R7=R5+R6
3400 R8=((R7*10)/100))/10
3500 R4(T)=((R3(T)+R8)*100)/255
3600 GOTO 500
3700 CLOSE
3750 OPEN "O",#1,"OXY.DAT"
3800 FOR I=1 TO T
3900 IF R4(I)=0 THEN 4100
4000 X(I)=R4(I)
4050 PRINT #1,I/6, X(I)
4100 NEXT I
4200 CLS
4300 SCREEN 2
4400 LINE (80,24)-(80,150),1
4500 LINE (80,150)-(500,150),1
4600 FOR K=1 TO 130 STEP 21
4700 LINE (75,23+K)-(80,23+K),1
4800 NEXT K
4900 FOR K2=1 TO 450 STEP 30
5000 LINE (79+K2,156)-(79+K2,151),1
5100 NEXT K2
5200 ON ERROR GOTO 5300
5300 FOR J=1 TO I
5400 X(J)=(X(J)*126)/60
5500 PSET (80+J,150-X(J)),1
5600 NEXT J
5700 ON ERROR GOTO 5800
5800 REM
5900 LOCATE 2,7:PRINT "Oxygen"
6000 LOCATE 3,7:PRINT 60
6100 LOCATE 19,7:PRINT 0
6200 LOCATE 21,10:PRINT 0
6300 LOCATE 21,17:PRINT 10
6400 LOCATE 21,24:PRINT 20
6500 LOCATE 21,32:PRINT 30
6600 LOCATE 21,39:PRINT 40
6700 LOCATE 21,47:PRINT 50
6800 LOCATE 21,54:PRINT 60
```

```
6900 LOCATE 21,62:PRINT 70
7000 LOCATE 22,32:PRINT "time (hours)"
7100 LOCATE 23,7:PRINT "Exp = ";B$
7200 INPUT D
7300 CHAIN  "ENTER.BAS"
7400 END
```

PHSUP.BAS

```
100 DIM R3(1500),R4(1500),X(1500),Y(1500)
200 OPEN "I",#1,"WORK.DAT"
300 INPUT #1,B$
400 KEY OFF
500 REM
600 T=T+1
700 INPUT #1,X
800 ON ERROR GOTO 3700
900 IF X>64 THEN 1200
1000 X=(X-48)*16
1100 GOTO 1300
1200 X=(X-55)*16
1300 R=X
1400 INPUT #1,X
1500 IF X>64 THEN 1800
1600 X=(X-48)
1700 GOTO 1900
1800 X=(X-55)
1900 R1=X
2000 R3(T)=R+R1
2100 INPUT #1,X
2200 IF X>64 THEN 2500
2300 X=(X-48)*16
2400 GOTO 2600
2500 X=(X-55)*16
2600 R5=X
2700 INPUT #1,X
2800 IF X>64 THEN 3100
2900 X=(X-48)
3000 GOTO 3200
3100 X=(X-55)
3200 R6=X
3300 R7=R5+R6
3400 R8=((R7*10)100))/10
3500 R4(T)=((R3(T)+R8)*14)/255
3600 GOTO 500
3700 CLOSE
3700 OPEN "O",#1,"PHT.DAT"
3800 FOR I=1 TO T
3900 IF R4(I)=0 THEN 4100
4000 X(I)=R4(I)
4050 PRINT #1,I/6,(X(I))
4100 NEXT I
4200 CLS
4300 SCREEN 2
```

```
4400 LINE (80,24)-(80,150),1
4500 LINE (80,150)-(500,150),1
4600 FOR K=1 TO 130 STEP 21
4700 LINE (75,23+K)-(80,23+K),1
4800 NEXT K
4900 FOR K2=1 TO 450 STEP 30
5000 LINE (79+K2,156)-(79+K2,151),1
5100 NEXT K2
5200 ON ERROR GOTO 5300
5300 FOR J=1 TO I
5400 X(J)=(X(J)*126)/60
5500 PSET (80+J,150-X(J)),1
5600 NEXT J
5700 ON ERROR GOTO 5800
5800 REM
5900 LOCATE 2,7:PRINT "pH"
6000 LOCATE 3,7:PRINT 60
6100 LOCATE 19,7:PRINT 0
6200 LOCATE 21,10:PRINT 0
6300 LOCATE 21,17:PRINT 10
6400 LOCATE 21,24:PRINT 20
6500 LOCATE 21,32:PRINT 30
6600 LOCATE 21,39:PRINT 40
6700 LOCATE 21,47:PRINT 50
6800 LOCATE 21,54:PRINT 60
6900 LOCATE 21,62:PRINT 70
7000 LOCATE 22,32:PRINT "time (hours)"
7100 LOCATE 23,7:PRINT "Exp = ";B$
7200 INPUT D
7300 CHAIN  "ENTER.BAS"
7400 END
```

BIOSUP.BAS

```
100 DIM R3(1500),R4(1500),X(1500),Y(1500
200 OPEN "I",#1,"WORK.DAT"
300 INPUT #1,B$
400 KEY OFF
500 REM
600 T=T+1
700 INPUT #1,X
800 ON ERROR GOTO 3700
900 IF X>64 THEN 1200
1000 X=(X-48)*16
1100 GOTO 1300
1200 X=(X-55)*16
1300 R=X
1400 INPUT #1,X
1500 IF X>64 THEN 1800
1600 X=(X-48)
1700 GOTO 1900
1800 X=(X-55)
1900 R1=X
2000 R3(T)=R+R1
2100 INPUT #1,X
2200 IF X>64 THEN 2500
```

```
2300 X=(X-48)*16
2400 GOTO 2600
2500 X=(X-55)*16
2600 R5=X
2700 INPUT #1,X
2800 IF X>64 THEN 3100
2900 X=(X-48)
3000 GOTO 3200
3100 X=(X-55)
3200 R6=X
3300 R7=R5+R6
3400 R8=((R7*10)/100))/10
3500 R4(T)=((R3(T)+R8)*100)/255
3600 GOTO 500
3700 CLOSE
3800 OPEN "O",#1,"BIOM.DAT"
3900 FOR I=1 TO T
4000 IF R4(I)=0 THEN 4500
4100 R4(I)=(LOG(100/R4(I)))/2.303
4200 X(I)=((R4(I)+0.001819)/0.0862035)/2.4
4300 PRINT I/6,X(I)
4400 PRINT #1,I/6,(X(I))
4500 NEXT I
4600 CLOSE
4700 STOP
4800 CLS
4900 SCREEN 2
5000 LINE (80,24)-(80,150),1
5100 LINE (80,150)-(500,150),1
5200 FOR K=1 TO 130 STEP 21
5300 LINE (75,23+K)-(80,23+K),1
5400 NEXT K
5500 FOR K2=1 TO 450 STEP 30
5600 LINE (79+K2,156)-(79+K2,151),1
5700 NEXT K2
5800 ON ERROR GOTO 5300
5900 FOR J=1 TO I
6000(J)=(X(J)*126)/60
6100 PSET (80+J,150-X(J)),1
6200 NEXT J
6300 ON ERROR GOTO 5800
6400 REM
6500 LOCATE 2,7:PRINT "bio(g/l)"
6600 LOCATE 3,7:PRINT 60
6700 LOCATE 19,7:PRINT 0
6800 LOCATE 21,10:PRINT 0
6900 LOCATE 21,17:PRINT 10
7000 LOCATE 21,24:PRINT 20
7100 LOCATE 21,32:PRINT 30
7200 LOCATE 21,39:PRINT 40
7300 LOCATE 21,47:PRINT 50
7400 LOCATE 21,54:PRINT 60
7500 LOCATE 21,62:PRINT 70
7600 LOCATE 22,32:PRINT "time (hours)"
7700 LOCATE 23,7:PRINT "Exp = ";B$
7800 INPUT D
7900 CHAIN  "ENTER.BAS"
8000 END
```

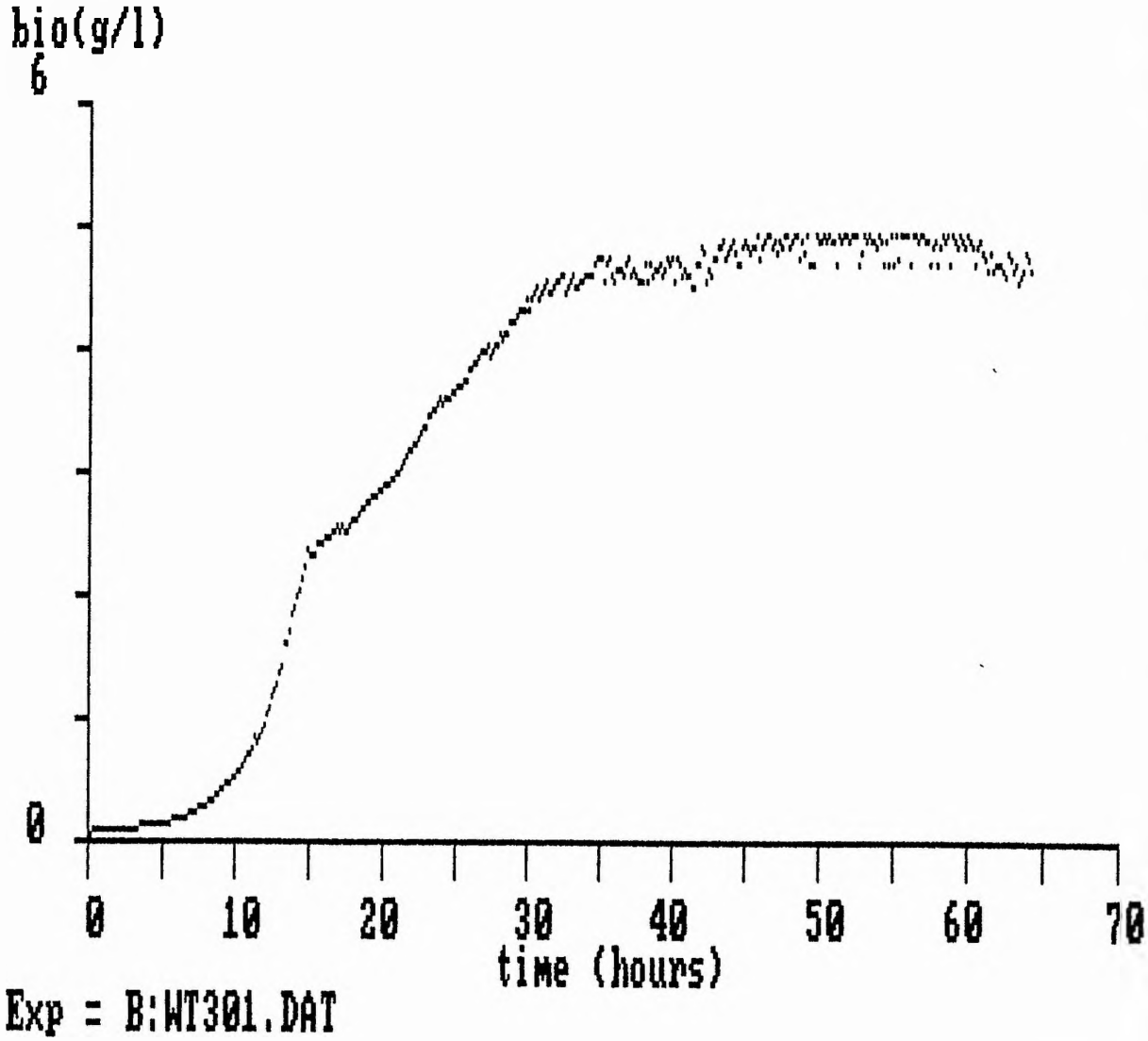


Figure 5.2.1.1 - Batch culture curve of Saccharomyces cerevisiae wild type Σ1278b using GRAPH1.bas and BIOSUP.BAS programs.

5.3 - Advanced Graphical and Statistical Analysis.

This package consist of 2 sets of programs in BASIC used to further process data from the PCS as follows:

5.3.1 Advanced Graphical Package.

5.3.2 Statistical Package.

5.3.1 - Advanced Graphical package.

This graphical package consists of two different options:

5.3.1.1 - Screen graphics.

The PLODISG.BAS (see APPENDIX D based on Anglle (215)), is much more advanced that the previous one (section 5.2.1 and 5.2.1.2), it can draw a XY plot, but also more information can be added such as the names of axes, names of parameters, text or draw other symbols or figures, etc.

5.3.1.2 - Plotter graphs (Epson HI-80 plotter).

The Plotter.bas (see APPENDIX E) program was designed to work with an Epson HI-80 plotter. It can draw a Y-time plot on a A4 paper using up to four different colours with fifteen different symbols. It is in BASIC without any special graphics functions. It

also can draw a simulate curve from a linear or polynomial equation written in the following way and incorporated into line 10300 of the program listed in appendix E.

Linear regression of an experiment.

slope = 0.060969; intercept = 3.114

Example: 10300 Y=3.114 + 0.060968 * (X)

5.3.2 - Statistical package.

As a statistical package a few programs from the STATPACK (Statistical package - NORTHWEST ANALYTICAL, Portland, OR, U.S.A.), were used. These included the program POLYREG.BAS, used to calculate regression coefficients from a straight line or a curve using a matrix algebra method, and CHIEQU.BAS and TTEST.BAS programs used to calculate Chi-square and T-Student test statistics for equal expected frequencies respectively given the values of a set of observed frequencies.

6 - RESULTS AND DISCUSSION.

6.1 - Fermenter System.

The fermenter system was designed to operate in batch or continuous culture using a standard PYREX glass vessel, fitted with a side port (1.5 litre) with a working capacity of 0.6 litre. The vessel has on its stainless steel top plate several input ports which were used to carry the various transducers (oxygen, temperature, pH, biomass), the heater and also tubes for input of medium and other compounds (alkali, acid, anti-foam)(figure 6.1). A BIOLAB agitator, operating at 200 rpm, was used to mix the culture. This speed was used as a control procedure to minimise the interference of bubbles on the operation of the biomass electrode.

The transducers used in this system were classified into two groups: Expensive and Inexpensive. The Expensive (over £ 60) group is represented by the oxygen and the pH steam-sterilisable sensors which are commercially available. The inexpensives group include the temperature and biomass probes and the system for measurement of alkali or acid addition to the fermenter. This group were constructed as part of this work.

6.1.1 - Wall Growth of Microorganisms.

A problem which can occur in fermentation is wall growth of microorganisms. They can adhere to glass and metal surfaces (217), especially in continuous culture of long

duration, where wall growth can vary from a light film of biomass to a massive accumulation on the vessel surface (120) and, in turbidimetric systems, microorganisms can grow preferentially on the detector (116). To overcome this problem of wall-growth, particularly on the biomass electrode, a teflon membrane was used to cover the top of the detector and the internal side of the glass wall which is part of the light pathway providing a permanent non-stick surface (195)(See biomass electrode description section 4.5.4.). No wall growth interference was detectable between the light pathway of the biomass electrode. This was checked by measuring biomass in the fermenter in 3 ways:

1 - Using the biomass electrode;
2 - Spectrophotometry on samples withdrawn;
3 - Dry weight estimations on samples withdrawn.

6.1.2 - Air Bubbles.

Small air bubbles, produced in great quantity, are unavoidable in vigorously-aerated culture vessels. A submersible colorimeter probe that permits easy and continuous measurement of optical density of media by removal of air bubbles has been developed by Ohashi et. al (120). It has an internal chamber through which the broth travels slowly causing interference on readings. In the PCS this interference has been reduced by:

1 - A suitable location of the detector inside the fermenter vessel (1 cm from the bottom and 0.5cm from the side

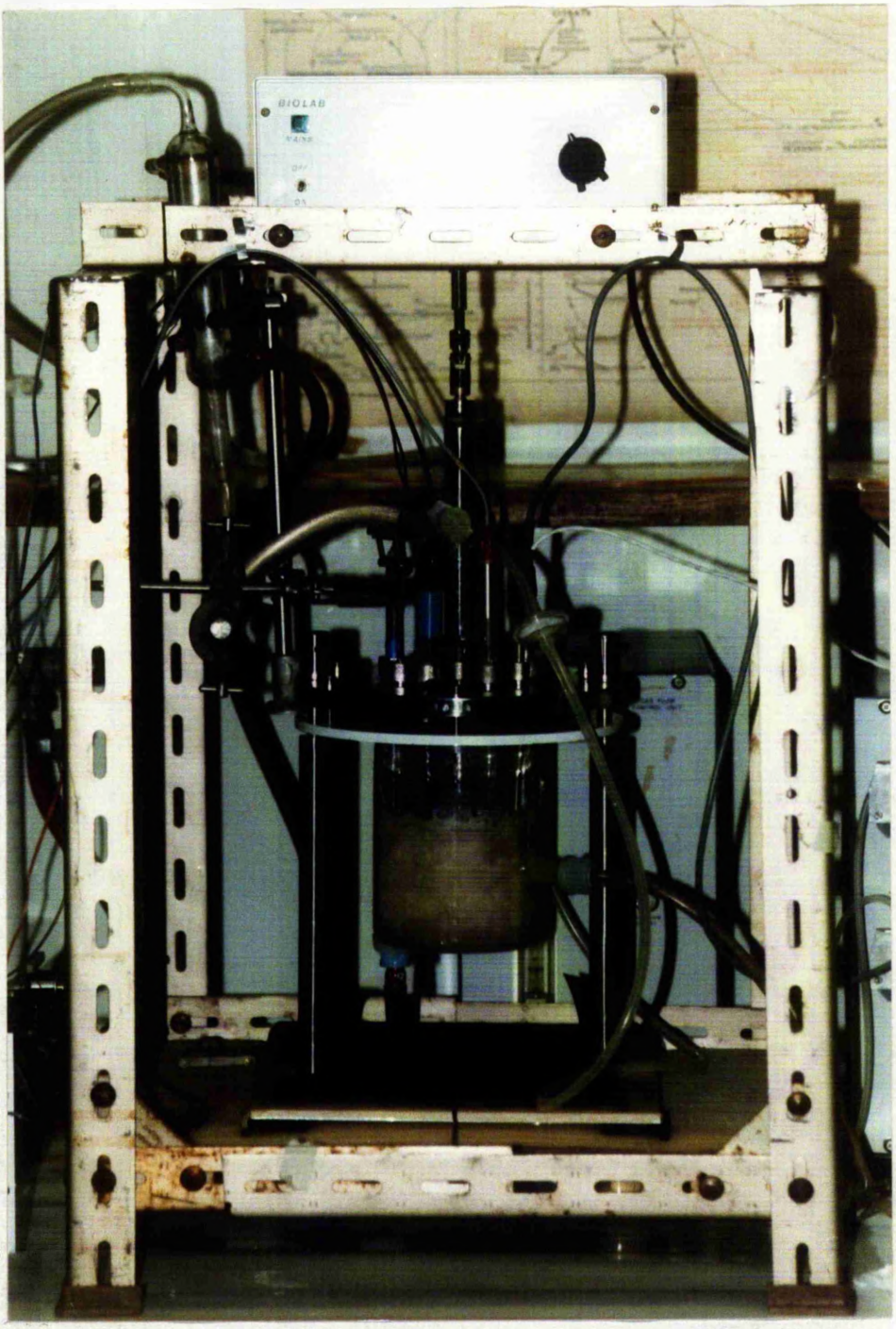


Figure 6.1 - Photograph of the PYREX glass vessel with its stainless steel top plate.

wall) (see biomass electrode description - section 4.5.4);

2 - Control of the light intensity and the rate of agitation at 200 rpm (sufficient to maintain constant the oxygen saturation, at 30%);

3 - Utilisation of a "software filter" (The averaging of parameter values)

6.1.3 - Foaming.

Foaming in continuous culture is a phenomenon that should be avoided as not only does it increase the chances of "inoculating" the medium inflow line but it can also interfere with the steady-state in several ways such as introducing another phase into the culture. Depending on the nature of the microorganisms, they can accumulate in the foam to a cell concentration much higher than that of the culture as a whole (198). To avoid interference by foaming in this system basically two procedures were carried out as follows:

1 - The medium composition was designed to produce a low level cell concentration;

2 - The stirring speed was maintained low (200 rpm).

However, in the case of foam interference, anti-foam

could be added to the fermenter using a tube connected to a reservoir containing anti-foam. Fortunately, addition of antifoam proved unnecessary.

6.2 - Processor Control System.

6.2.1 - Hardware of the PCS.

The Hardware used in the PCS is simple and inexpensive to build. The system has several advantages from the academic point of view as follows:

6.2.1.1 - Advantages.

A. The components used in this system are inexpensive and easy to obtain in developing countries.

B. This PCS allows good access to the circuit boards (located in an open rack framework) and being open it needs no fan cooling.

C. The wire-wrapping technique, used in construction, is flexible and allows easy circuit modification.

D. The PCS has two serial links (RS 232C) for communication with other computers.

E. The Memory board has plenty of space for expansion, especially the EPROM system, which could be doubled in size.

F. The Analog/digital and ON/OFF Switch board also have a lot of space, and, due to the software architectural design, the number of analog/digital channels available can be expanded. The Analog/digital conversion already includes a multiplexer which minimises undesirable noise in the conversion caused by extra circuits.

A early version of the PCS had 2 serial ports used to communicate with another computer and one parallel port used to communicate with a printer. This latter port was never used due to operational difficulty. In this version only the 2 serial ports have been utilised.

6.2.2 - Software of the PCS.

The software used in this system has both advantages and disadvantages.

6.2.2.1 - Advantages.

6.2.2.1.1 - Language and Loop system.

The main advantage of the software in the PCS is the programming language used and the small size of the main program. The assembler language uses the machine code of the Z-80A microprocessor which has one of the most versatile code libraries of 8-bit microprocessors, consequently the software is more simple than that of other 8-bit microprocessors

(8080, 8085).

The loop system, used for data acquisition and control, was controlled by a Real Time Clock (RTC) system which uses a very simple program (see Appendix C). Consequently, it is very fast in execution, avoiding problems of delays between data acquisition and control.

Due to the high speed and efficiency of the CONTROL loop (see sections 5.1.2 and 5.1.3) which controls the ON/OFF Switch-circuit (used to control conditions in the fermenter) a digital/analog converter was not used.

6.2.2.1.2 - Noise reduction.

To reduce noise during data acquisition and control caused by electrical interference 4 approaches have been applied:

1 - The use of the data acquisition system in a environment with as little electrical interference as possible.

2 - Length of cable from sensor to interface board should be approximately 1 meter (this is especially important for the pH probe).

3 - Electronic filters using opto circuits for data transmission should be employed.

4 - The use of a subroutine (software filter) which

determines an average value from 100 points from each parameter.

6.2.2.1.3 - Terminal display.

Information on the growth behaviour of a microbial culture is useful and desirable. This system uses two different procedures for this purpose:

1 - The vdu connected with the PCS continuously displays a Y-time graph (Y = a specific parameter, normally biomass, updated every 10 mins). This procedure allows, for instance, the growth curve of a microorganism to be displayed during batch growth. (see section 6.2.1.2 - Graphics quality).

2 - The second microcomputer (IBM - PCXT), can also show a graph with better resolution. However, as the IBM-PCXT operates at a second level in this hierarchial system, the graphic facilities on it can be used only at certain stages in an experiment (e.g. after a 'down-loading' of data from PCS memory).

6.2.2.1.4 - Flexibility of the PCS Software.

An important aspect of the PCS software system is its flexibility in allowing modifications using the extra available RAM memory. This software is thus in contrast to "black box" software where modification is impossible.

6.2.2.2 - Disadvantages of the PCS Software.

6.2.2.2.1 - Data Transfer.

Data transfer, while effective, suffers from the disadvantage that there is no "common memory software" (as in larger systems) allowing 2 CPUs the use of the same memory space. Thus, data transfer from the PCS to the second computer requires that sensor data acquisition (by the PCS) is temporarily halted. Subroutines D or L (see section 5.1.4) are then used to effect transfer which may take 15-30 seconds. The PCS RAM memory is big enough, however, to store data from an experiment for at least 4 days (10 min intervals).

6.2.2.2.2 - Graphics quality.

Unfortunately, the Hazeltine Esprit II terminal used with the PCS did not have any sophisticated graphics facilities, However, it was possible to obtain a rough, low resolution graphical representation (see figure 6.2.2.1.2).

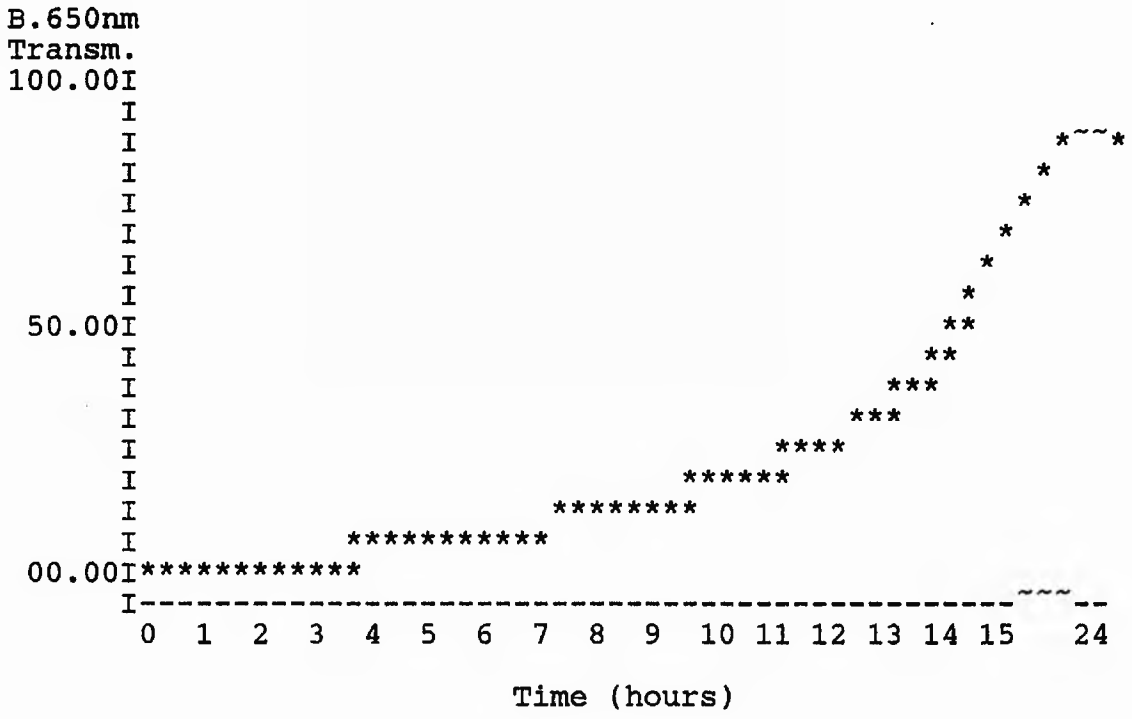


Figure 6.2.2.1.1 - Biomass curve of a batch experiment displayed on Hazeltine Esprit II terminal connected with the PCS

6.2.2.2.3. - Alarm signal.

An alarm signal could be produced by a program which checks that data parameter values from the fermenter within a specific range.

6.3 - Batch Culture Experiments.

6.3.1. Growth rate

Table 6.3.1 shows the maximum specific growth rate and the NADP-GDH and the GOGAT activities of three different strains of Saccharomyces cerevisiae, whose growth curves are shown in the figure 6.3.1. Two of them are recombinant DNA strains and one is a wild type strain used as control. The specific growth rates of the two recombinant DNA strains were below that of the wild type (table 6.3.1.). This result is in agreement with Lee et. al. (218), who showed that cells without plasmids were able to grow faster than those carrying plasmids. This is also in accordance with previous observations in bacteria (219, 183). Such behaviour might result from the added reproductive and metabolic loads on cells carrying extra plasmids (218). Furthermore, Srienc et. al. (186) have described experiments in which plasmid-bearing microorganisms grow more slowly than cells without plasmids in the absence of selection pressure. In addition, this growth rate penalty increases as the number of plasmids per cell increases and as the level of plasmid gene expression increases.

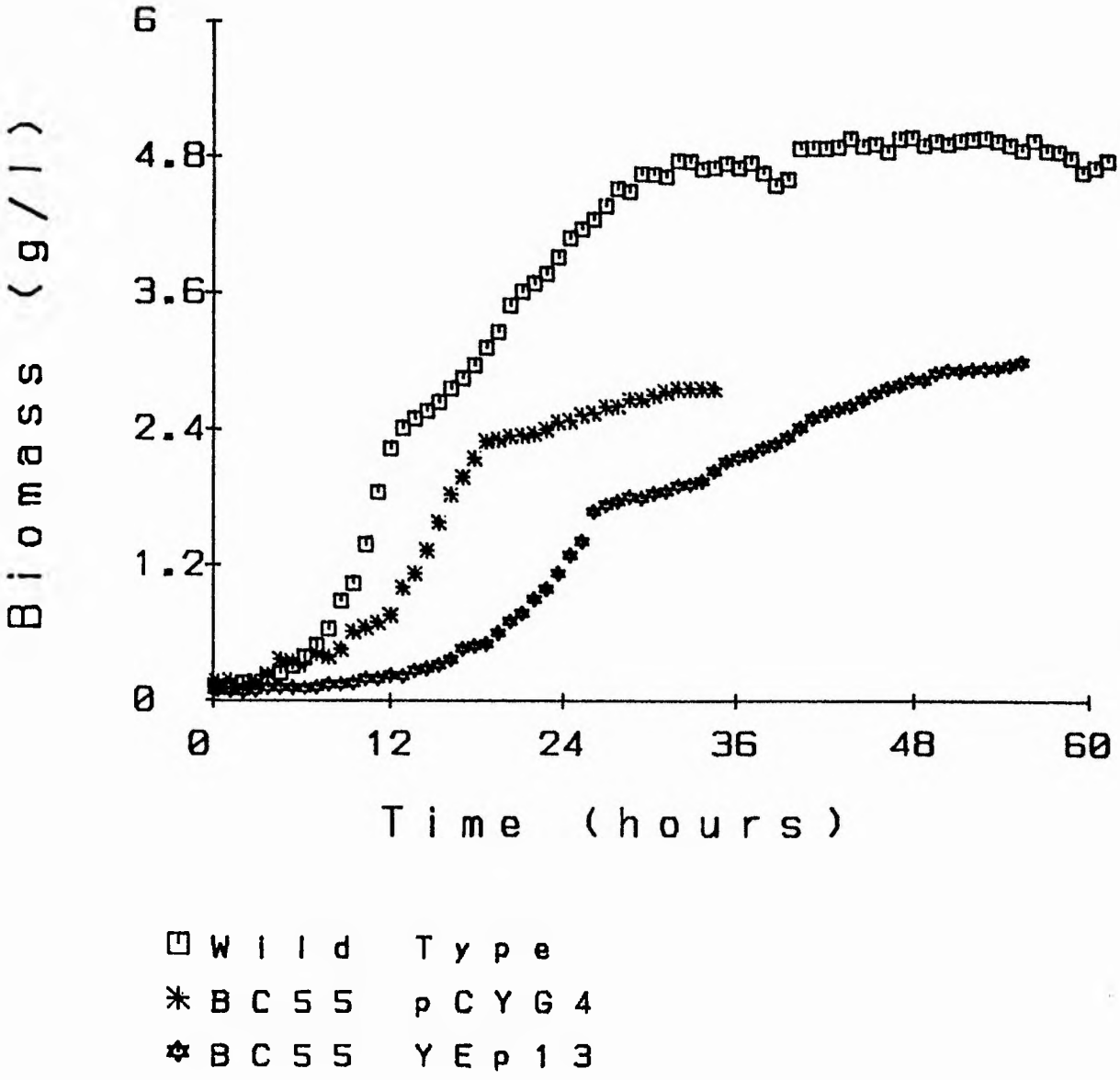


Figure 6.3.1. - Growth Curves of three different strains of Saccharomyces cerevisiae.

Table 6.3.1. Batch culture of three strains of *Saccharomyces cerevisiae*

Strain	Vectors	μmax (/h)	NADP-GDH U/mg protein	GOGAT U/mg protein
W. Type Σ1278b	------	0.286	0.474	0.190
BC55	pCYG4	0.248	5.391	1.090
BC55	YEp13	0.105	<0.01	0.230

Plasmid-encoded gene products may be proteins which are part of the normal metabolic pathways, such as NADP-GDH (the major pathway for ammonia assimilation in Saccharomyces cerevisiae) (159, 158). Lack of NADP-GDH in Saccharomyces cerevisiae can cause a reduction in growth rate on ammonia, since, in the absence of this pathway, cells start to utilise a more energetically expensive pathway (GOGAT pathway: a pathway with a high affinity for ammonia, but which utilises more ATP than the NADP-GDH pathway). Saccharomyces cerevisiae (strain BC55), lacking NADP-GDH (gdh^-), can overcome this problem using NADP-GDH coded by a NADP-GDH gene on the pCYG4 plasmid (11-fold more activity than NADP-GDH in Saccharomyces cerevisiae wild type Σ1278b). This plasmid also carries the gene for penicillinase (useful marker) and for leucine production (leucine is a selective pressure for BC55 cells (Leu^-)).

The slow growth of BC55 carrying the plasmid pCYG4 (gdh^+) compared with wild type cells (Σ1278b) can be related to extra plasmid protein production (e.g. penicillinase, enzymes involved in leucine biosynthesis) (218, 185).

BC55 cells without NADP-GDH activity carrying the plasmid YEp13 (same plasmid as pCYG4, but without NADP-GDH gene encoded) had a lower growth rate than both gdh^+ and wild type Σ1278b, consequential on the lack of NADP-GDH activity and the extra plasmid protein production (e.g. penicillinase, enzymes involved in leucine biosynthesis) (220, 218, 185).

6.3.2 - Glucose yield and ethanol production.

The glucose yield in gdh^+ cells was slightly greater than in gdh^- and wild type cells (table 6.3.2). Ethanol production was proportional to growth in all strains, but a greater ethanol production was found in the wild type cells, which could be consequential on the presence of plasmid (gdh^+ and gdh^- cells) since part of the glucose has been utilised in extra protein production.

Wild type cells can utilise ethanol as a carbon source better than gdh^+, this is based on the observation of the second phase (figure 6.3.1) of the growth curve of the cells, where ethanol has been used as a second carbon source (see figures 6.3.2a, 6.3.2b. and 6.3.2c.). The cells carrying either plasmid had a slower growth rate compared to wild type cells when ethanol was used as carbon source.

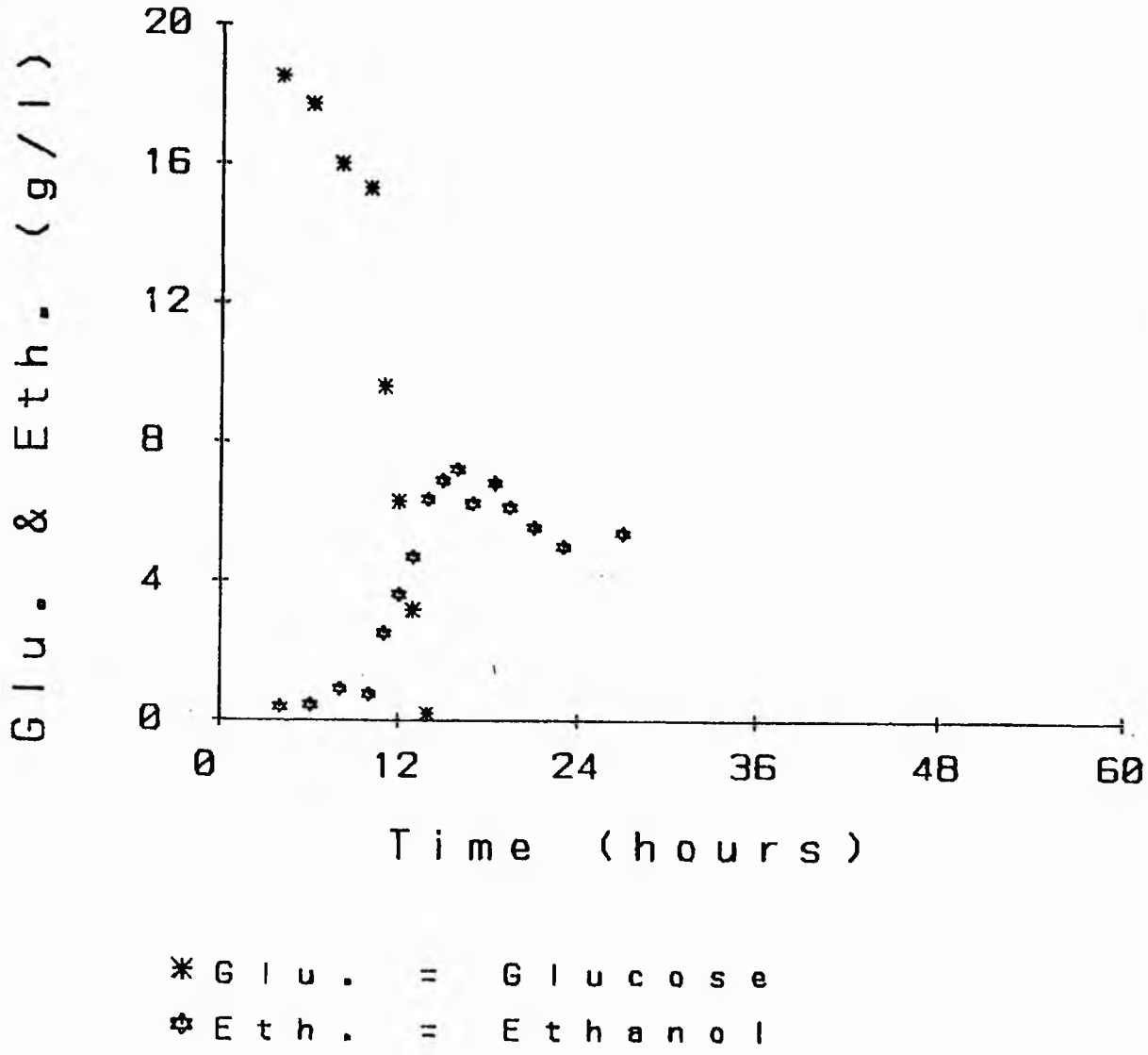


Figure 6.3.2a. - Curves of glucose utilisation and ethanol production of Saccharomyces cerevisiae Wild type Σ1278b.

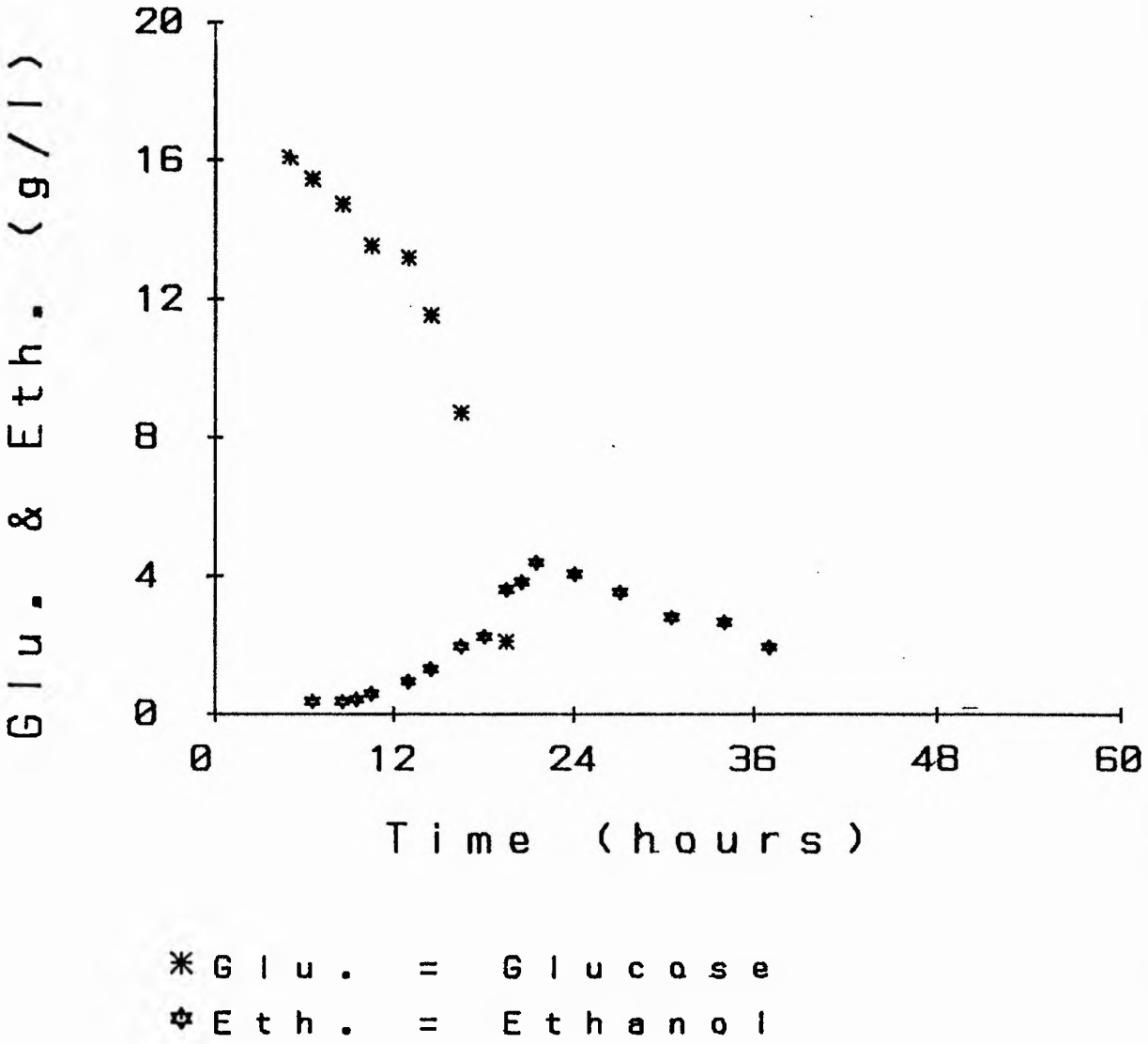


Figure 6.3.2b. - Curves of glucose utilisation and ethanol production of Saccharomyces cerevisiae, strain BC55 carrying the plasmid pCYG4 (gdh^+).

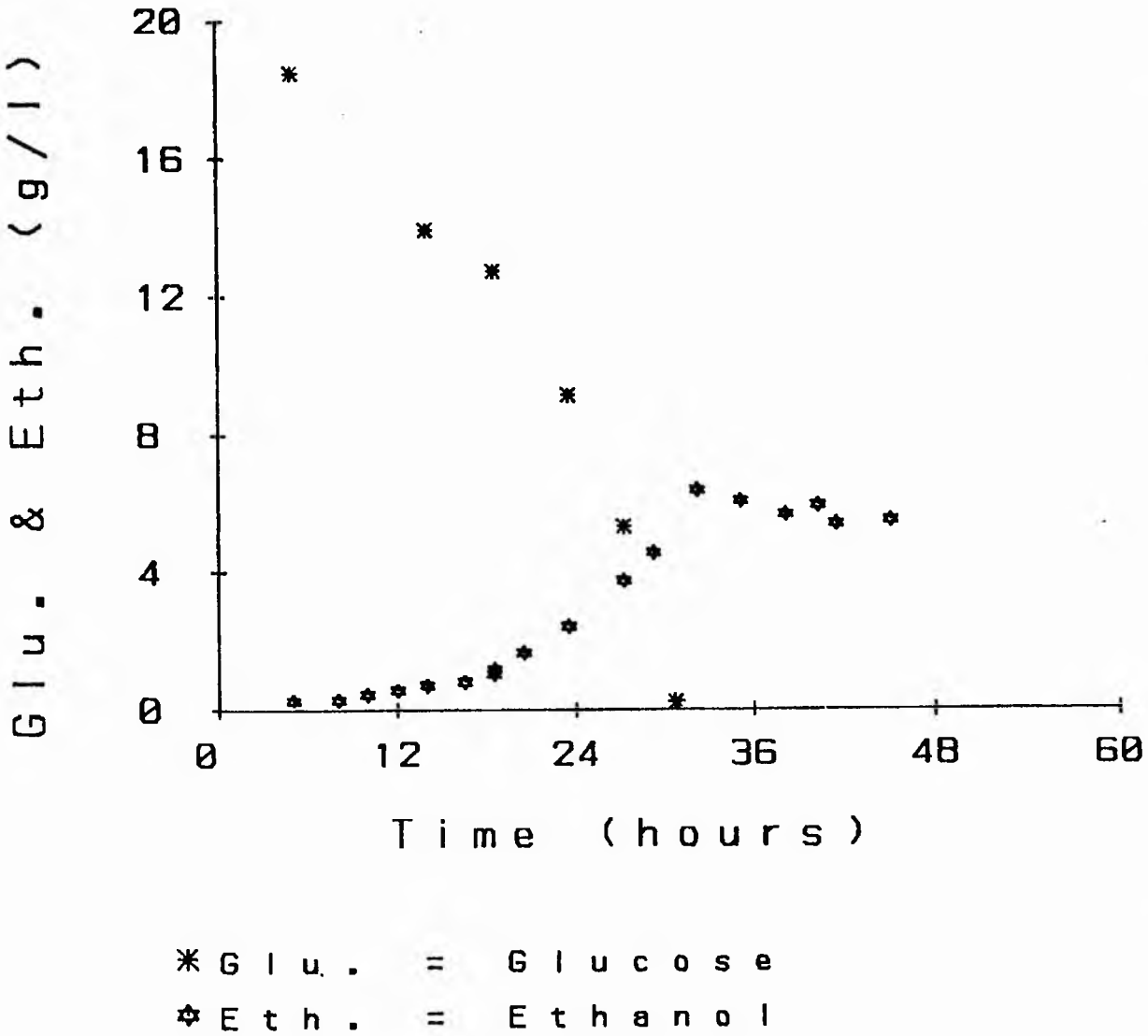


Figure 6.3.2c. - Curves of glucose utilisation and ethanol production of Saccharomyces cerevisiae, strain BC55 carrying the plasmid YEp13 (gdh⁻).

Table 6.3.1.1. Batch culture of three strains of *Saccharomyces cerevisiae*

Strain	Vectors	μmax (/h)	Glucose Yield*	Ethanol Production rate (g/l/h)	Ethanol Production maximum (g/l)
Wild Type Σ1278b	-----	0.286	0.128	0.299	7.25
BC55	pCYG4	0.248	0.151	0.238	4.37
BC55	YEp13	0.105	0.125	0.097	6.37

* Gram biomass/ Gram of glucose used.

6.3.3 - Comparative levels of GOGAT activity.

Table 6.3.1 shows a 5-fold increase in the GOGAT activity which is related with an increase in NADP-GDH activity in gdh^+ cells. Roon et. al. (160), pointed out that variations in the specific activity of GOGAT are generally parallel to NADP-GDH activity. However, the maximum level of activity for both GDHs (NADP and NAD dependent) has been shown (160) to be more than 10 fold higher than the maximal level of GOGAT activity, which differs from the present results (table 6.3.1) which show a 5 fold difference between NADP-GDH and GOGAT activities. However, Roon et. al. (160) suggested that GOGAT serves in a capacity which is auxiliary to the NADP-GDH, perhaps as a means for directly converting the amide group of glutamine to α-amino nitrogen without contributing to the pool of free ammonia. Consequently, the increases of NADP-GDH could stimulate the increase in GOGAT.

6.3.4 - Comparative levels of NADP-GDH activity.

The NADP-GDH gene encoded in the vector YEp13 (plasmid pCYG4) confers an 11 fold increase in the activity of this enzyme in the BC55 cells (table 6.3.1) comparing with wild type cells. Similar results, but with a 18 fold increase were found by Nagasu and Hall (149), and a 10 fold increase, by Moye et. al. (150), using both the same Escherichia coli-yeast shuttle vector YEp13 to encode the NADP-GDH gene.

However, Moye et. al. (185) have shown that despite the 10 fold increase in NADP-GDH, GOGAT activity was unchanged in cells with plasmid and those without plasmid, which differs from the present results, where the GOGAT activity was almost five fold higher than wild type (table 6.3.1), however, Moye et. al. (185) used a strain which has NADP-GDH in the genome, which could be responsible for some repressive control on the GOGAT gene.

6.3.5 - Uptake of ammonia.

Protons released during the ammonia uptake process cause the pH to decrease (221). Wiame et. al. (222) have suggested that, during synthesis of a dipeptide from glucose and ammonia, there is a concomitant production of one proton equivalent for each nitrogen atom assimilated, following the equation below:

$$C_6H_{12}O_6 + 2NH_4 \dashrightarrow C_6O_3N_2H_{12} + 3H_2O + 2H^-$$

San and Stephanopoulos (221) grew Saccharomyces cerevisiae controlling the pH of the culture by the periodic addition of ammonium hydroxide. They observed that the measurements of ammonium hydroxide added to the reactor could be used to estimate:

1 - Biomass;
2 - Substrate consumption;
3 - Product production;
4 - Yields.

By using the system described in section 4.5.6, the amount of ammonium hydroxide (2M) used to maintain the pH constant inside the fermenter vessel was measured (Figure 6.3.5.).

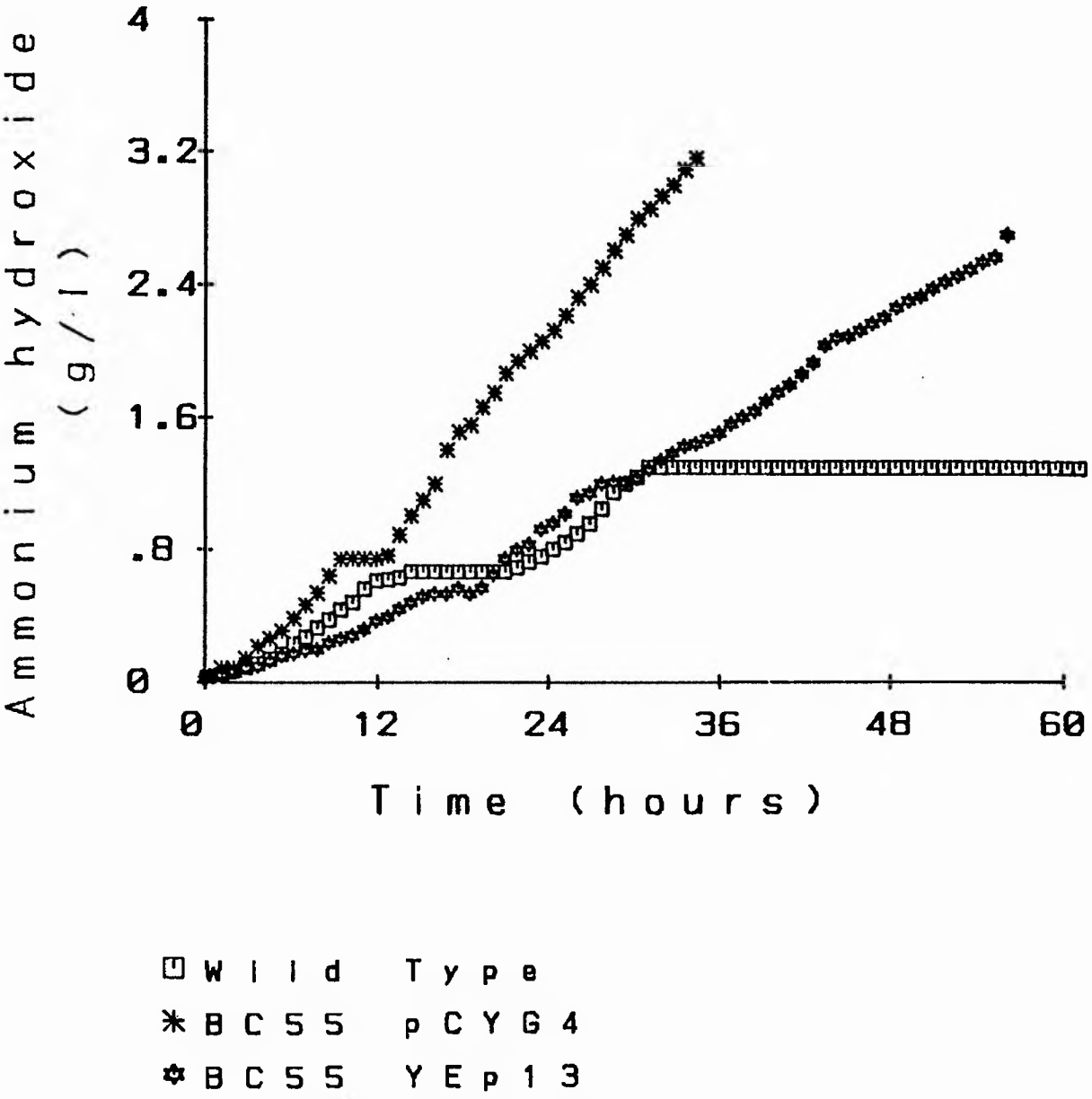


Figure 6.3.5. - Ammonium hydroxide used to maintain constant pH inside the fermenter vessel (three different strains of Saccharomyces cerevisiae).

The ammonia uptake rate in wild type and gdh^+ cells was the same and higher than in gdh^- cells when glucose was used as carbon source (figures 6.3.5a, 6.3.5b and 6.3.5c). Furthermore, the wild type growth curve shows 4 phases related to pH changes (figure 6.3.5a) which differs from the growth curves of gdh^+ and gdh^- cells (2 phases). The first phase of the wild type cells corresponds to the uptake of glucose. The growth rate in this phase is rapid and the metabolism is primarily glycolytic resulting in the accumulation of ethanol in the medium (222), then a plateau which corresponds to the diauxic lag (a change to ethanol metabolism when glucose is exhausted in the medium), a period during which the enzymes necessary for respiratory growth became rapidly derepressed, while a corresponding decline was observed in the activities of enzymes associated with glycolysis and ethanol fermentation (223). The other exponential phase occurs when ethanol is used as a second carbon source. Eventually, when substrates are exhausted in the culture, the pH rises due to the increase in K^+ and decrease in H^+ concentration in the medium (224).

In batch culture experiments with gdh^+ and gdh^- cells the culture pH value did not increase (figures 6.3.5b and 6.3.5c) which differs from the wild type Σ1278b (figure 6.3.5a) which showed 2 peaks in pH the first in phase with the diauxic lag, the second when the substrate was exhausted in the medium. However, the amount of ammonium hydroxide added was almost 2.5 times higher in gdh^+ cells than with wild type ones and 3 times higher than with gdh^- cells.

The lack of any plateau between the end of glucose utilisation and the start of ethanol utilisation (see figure 6.3.5b and 6.3.5c) suggests the presence of a very short diauxic lag with these cells (gdh^+ and gdh^-) compared with wild type ones.

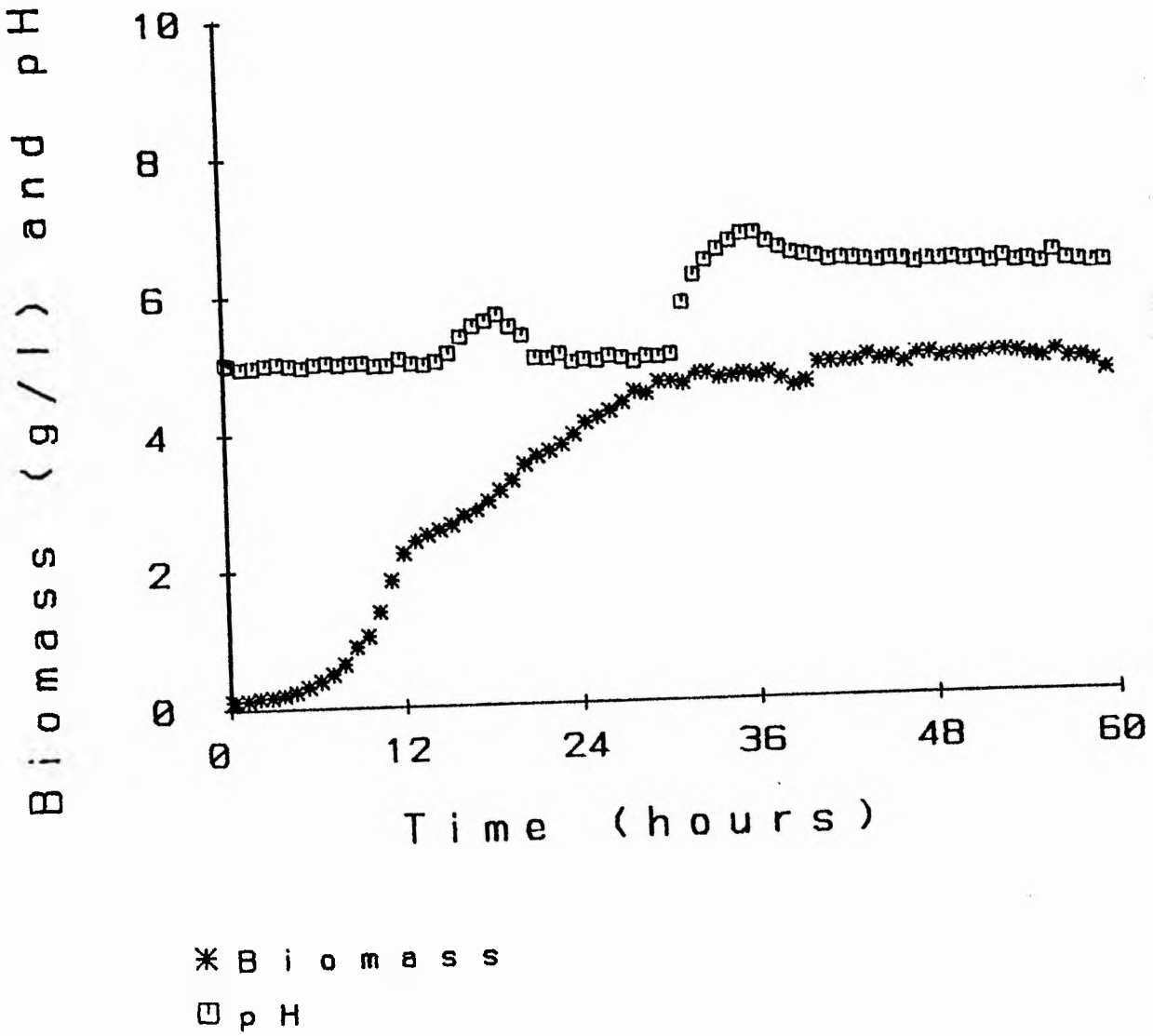


Figure 6.3.5a. - Growth and pH curve of Saccharomyces cerevisiae wild type Σ1287b, in batch culture.

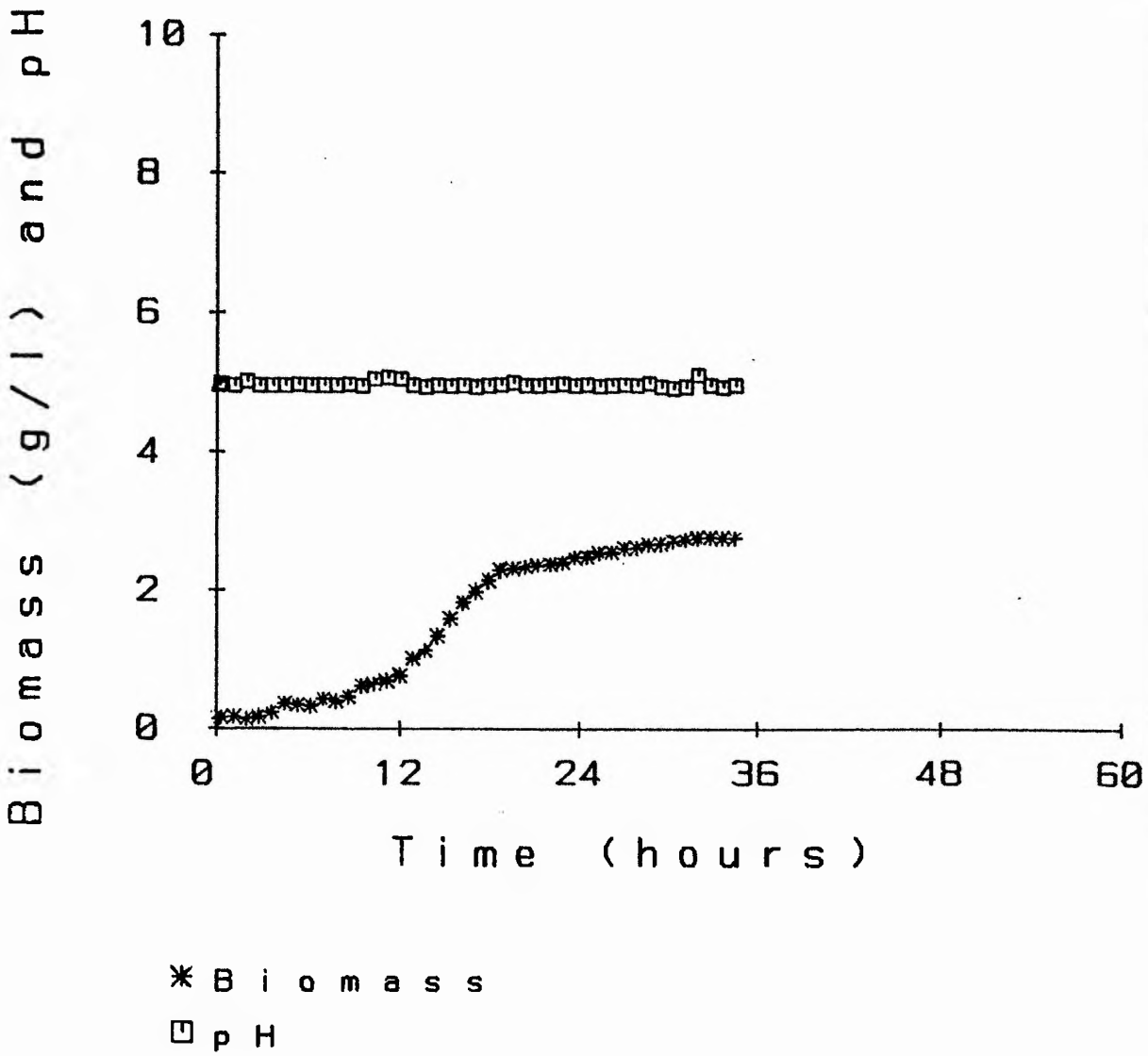


Figure 6.3.5b. - Growth and pH curve of Saccharomyces cerevisiae BC55 pCYG4 (gdh^+), in batch culture.

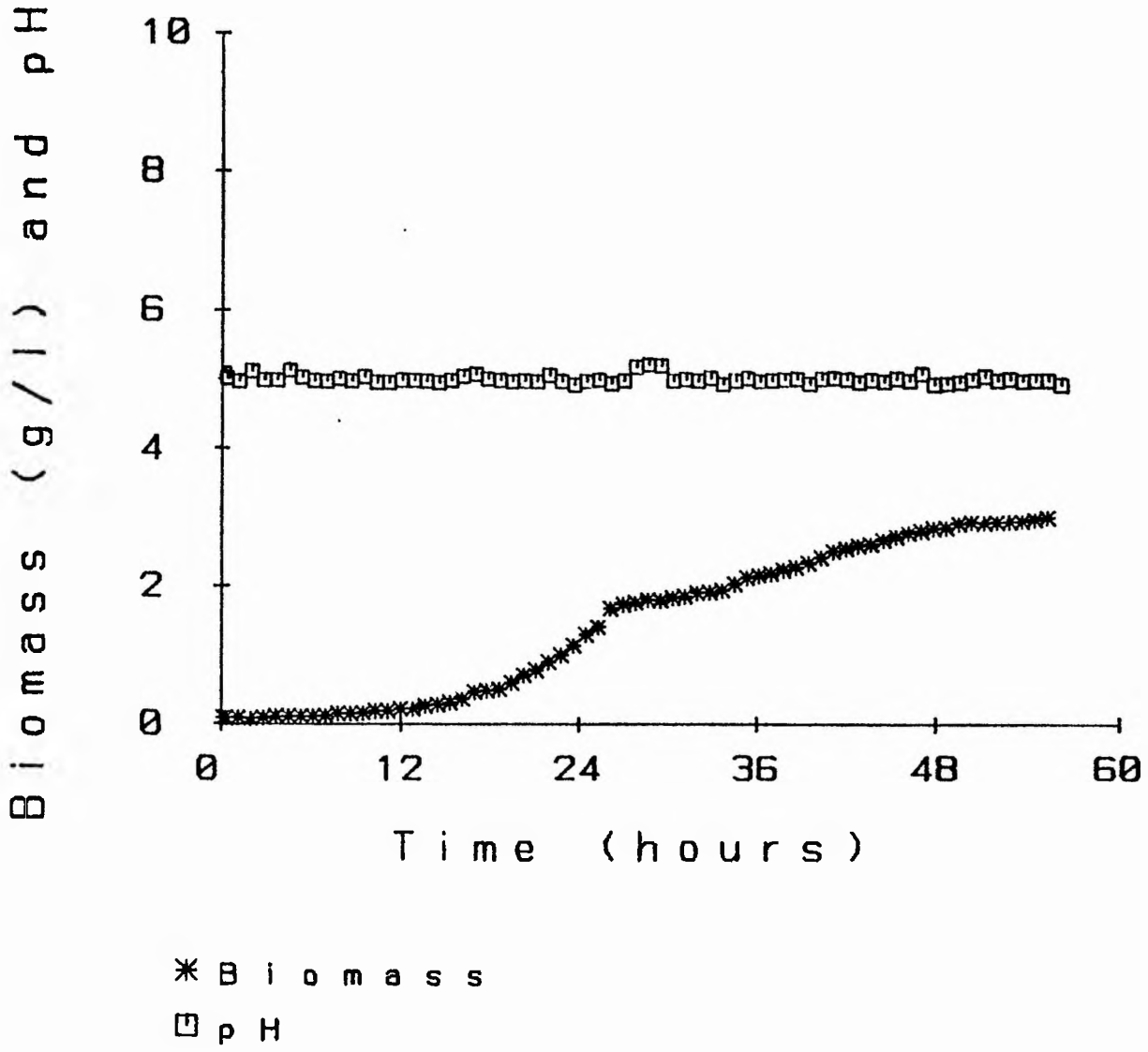


Figure 6.3.5b. - Growth and pH curve of Saccharomyces cerevisiae BC55 YEp13 (gdh^-), in batch culture.

Table 6.3.5 - Uptake of ammonia by three different *Saccharomyces cerevisiae* strains.

Strains	Vector	NH_4OH Uptake mM/h	NADP-GDH (U/mg protein)	Generation time (H)
Σ1278b	-----	.380	.4740	2.42
BC55	pCYG4	.360	5.391	2.79
BC55	YEp13	.110	<0.01	6.48

Furthermore, the ammonia transport in yeast is mediated by two active transport systems that operate over a narrow pH range (5.5-7.5) with maximal activity at pH 6.5 (224, 225). However Bogonez et. al. (153) have pointed out that at higher pH values ammonia uptake is essentially a diffusion process and that prevention of the pH increase in the medium resulted in a 6-fold decrease in intracellular ammonia concentration and that this decrease was accompanied by an increase in NADP-GDH activity from 185 to 500 nmol min^{-1} (mg protein)$^{-1}$. In addition, San and Stephanopoulos (221) pointed out that the amount of ammonia added to restore the pH to its set point is exactly equal to the amount taken up by the yeast cells during the growth. The present results suggest that the presence of the plasmid pCYG4 increases the amount of ammonia taken up by the cells, but does not increase the amount of biomass which suggests that, the extra ammonia has been used in the biosynthesis of plasmid gene products. Furthermore, considering the growth properties of a population containing plasmid-free cells(P^-) and plasmid-containing cells (P^+), the growth-rate penalty paid by plasmid-containing cells for synthesis of plasmids and plasmid products must be taken into consideration (150).

6.4 - Continuous Culture Experiments

All plot curves displayed in this section (best fitted plot curve ($p < 0.05$)) were determined by using the POLYREG (polinomial regression), CHIEQU (Chi-square test) and the TTEST (t-student test) programs (see section 5.3.2).

6.4.1 - Carbon as limiting substrate.

6.4.1.1 - Estimation of maximum specific growth rate of Saccharomyces cerevisiae BC55 (pCYG4) (gdh^+)

Maximum specific growth rates for gdh^+ cells were determined as indicated in table 6.4.1.1, both in batch mode (at chemostat startup) and, following maintenance of steady state for approximately 45 generations, by washout kinetics (section 3.2.2.5). It is apparent that there is no significant difference between the values under (a) and (b) for any intervening steady state growth rate. This indicates that, under carbon limitation between these steady state values, these is no observable change (as far as μ is concerned) in the cell population.

Table 6.4.1.1. - Maximum specific growth rates of gdh^+ cells, determined during chemostatic experiments (under carbon-limitation) , as follows:

(a) During exponential batch growth (chemostat startup).

(b) By washout kinetics from 4 different steady states.

Experiment	(a) μmax (batch h^-)	Steady State growth rate	(b) μmax (washout kinetics)
1	.248	.016	.247
2	.252	.044	.252
3	.264	.100	.246
4	.257	.150	.255

6.4.1.2 - NADP-GDH and GOGAT activities.

At four different dilution rates NADP-GDH activity increased proportionally to the dilution rate (figure 6.4.1.2a), with a maximum activity at 0.15/h, in agreement with Roon and Even (165) who reported that the specific activity of this enzyme is highest during exponential growth. Caulcott (185) also observed that the activity of NADP-GDH in yeast increases with increasing dilution rate, and, in carbon-limited culture, NADP-GDH reaches maximal activity when the dilution rate is approximately 0.15/h (170).

However, in the four experiments, GOGAT activity was not proportional to the dilution rate (figure 6.4.1.2b) and was lower than in experiments using ammonia as growth limiting substrate, since GOGAT does not have a high activity in the presence of high concentrations of ammonia. Furthermore, Stadtman and Ginsburg (226) and Magasanik (154) pointed out that in microorganisms having both aminating pathways (NADP-GDH and GOGAT), the GOGAT pathway is drastically decreased upon an increase in ammonia supply, and amination proceeds via L-glutamate dehydrogenase, since the GOGAT pathway has a minor importance in ammonia assimilation (159, 161).

GOGAT activity curves were parallel to NADP-GDH ones and were almost 10-fold lower than the maximum level of NADP-GDH activity which is in agreement with Roon et. al.,

(160) who have observed that variations in the specific activity of GOGAT, which occur in growing yeast cultures in response to changes in the nitrogen source, are generally parallel to changes observed with the NADP-GDH and that levels of GOGAT are approximately 10-fold lower than the maximum level of activity for NADP-GDH.

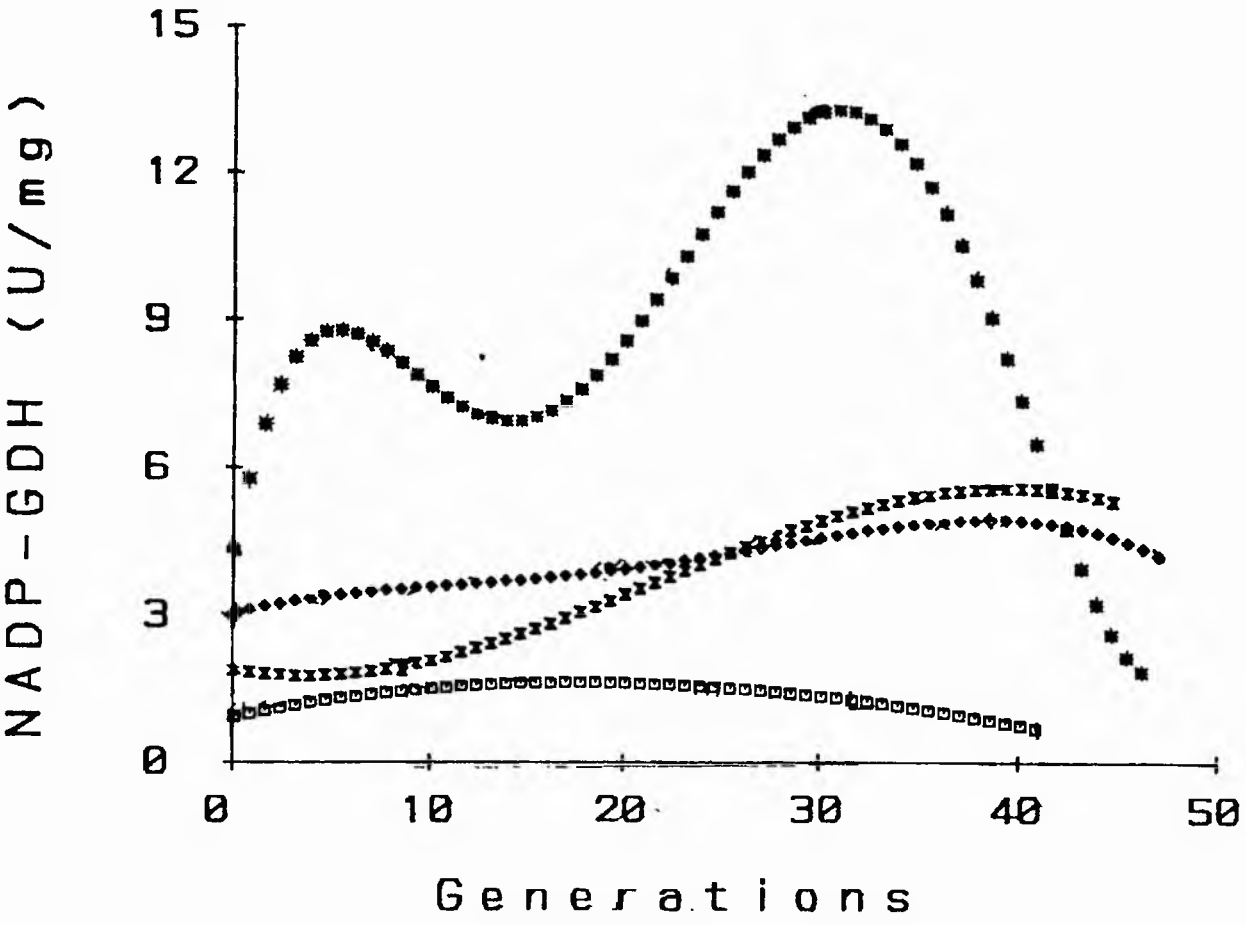


Figure 6.4.1.2a - NADP-GDH activity of gdh^+ cells growing under carbon-limited continuous culture.

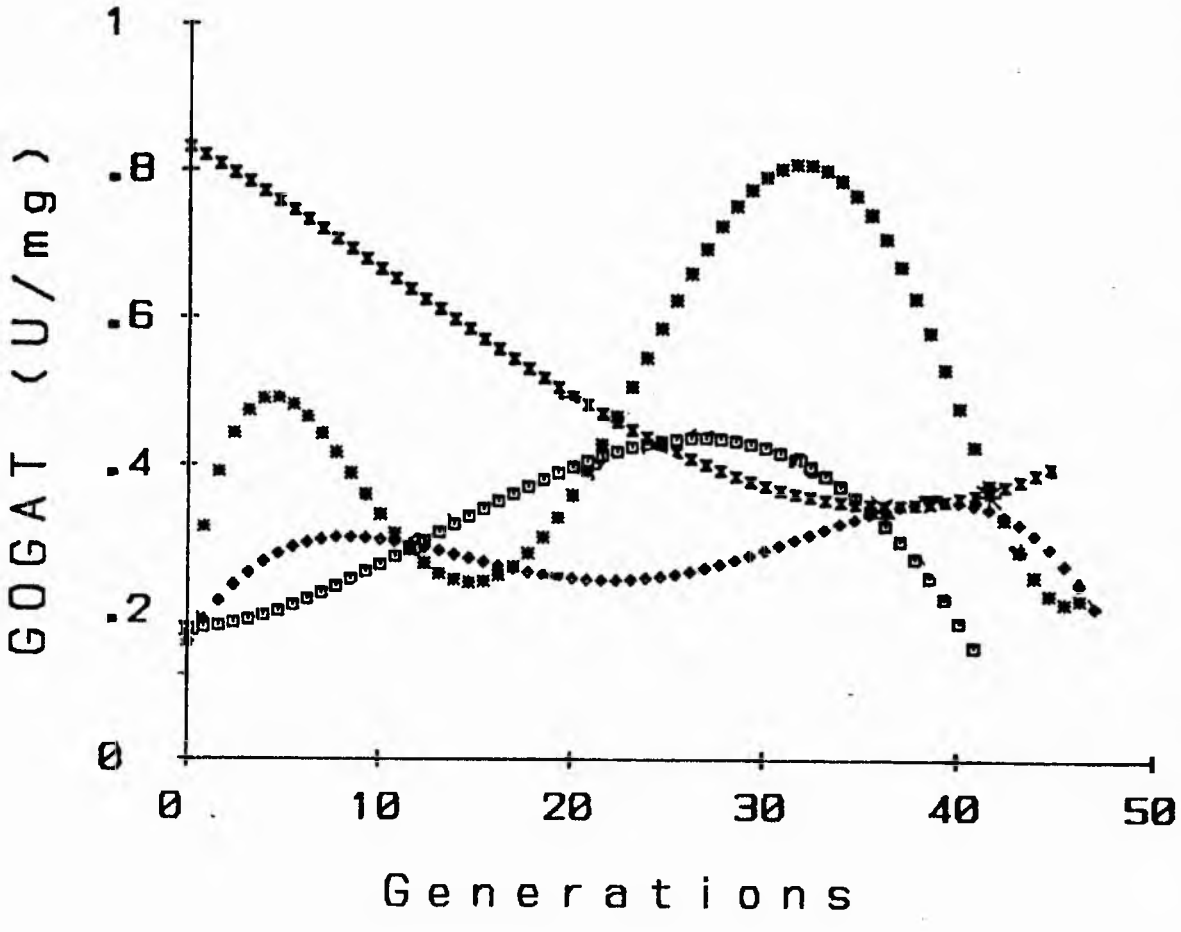


Figure 6.4.1.2b - GOGAT activity of gdh^+ cells in carbon-limited continuous culture.

6.4.1.3 - Intracellular concentration of ammonia and L-glutamate.

In these experiments the intracellular concentration of ammonia (figure 6.4.1.3a) was inversely related to the NADP-GDH and GOGAT activities. These results are in agreement with Bogonez et. al. (153) who observed a decrease of 2 and 7 fold of NADP-GDH activity with an increase of 10 and 20 fold ammonia content of the cells. Unlike the degradation process which NADP-GDH undergoes during carbon starvation (227), during ammonia accumulation there is merely a repression of synthesis, with small, if any, degradation. In contrast, L-glutamate concentration was proportional to both enzyme activities (NADP-GDH and GOGAT) (figure 6.4.1.3b), but not correlated to growth rate. In contrast, Bogonez et. al. (153) observed a decrease in NADP-GDH activity associated with a rise in internal ammonia or L-glutamate. However, Bossinger et. al. (228) pointed out that the extent of repression cannot be correlated with the culture ammonia concentration.

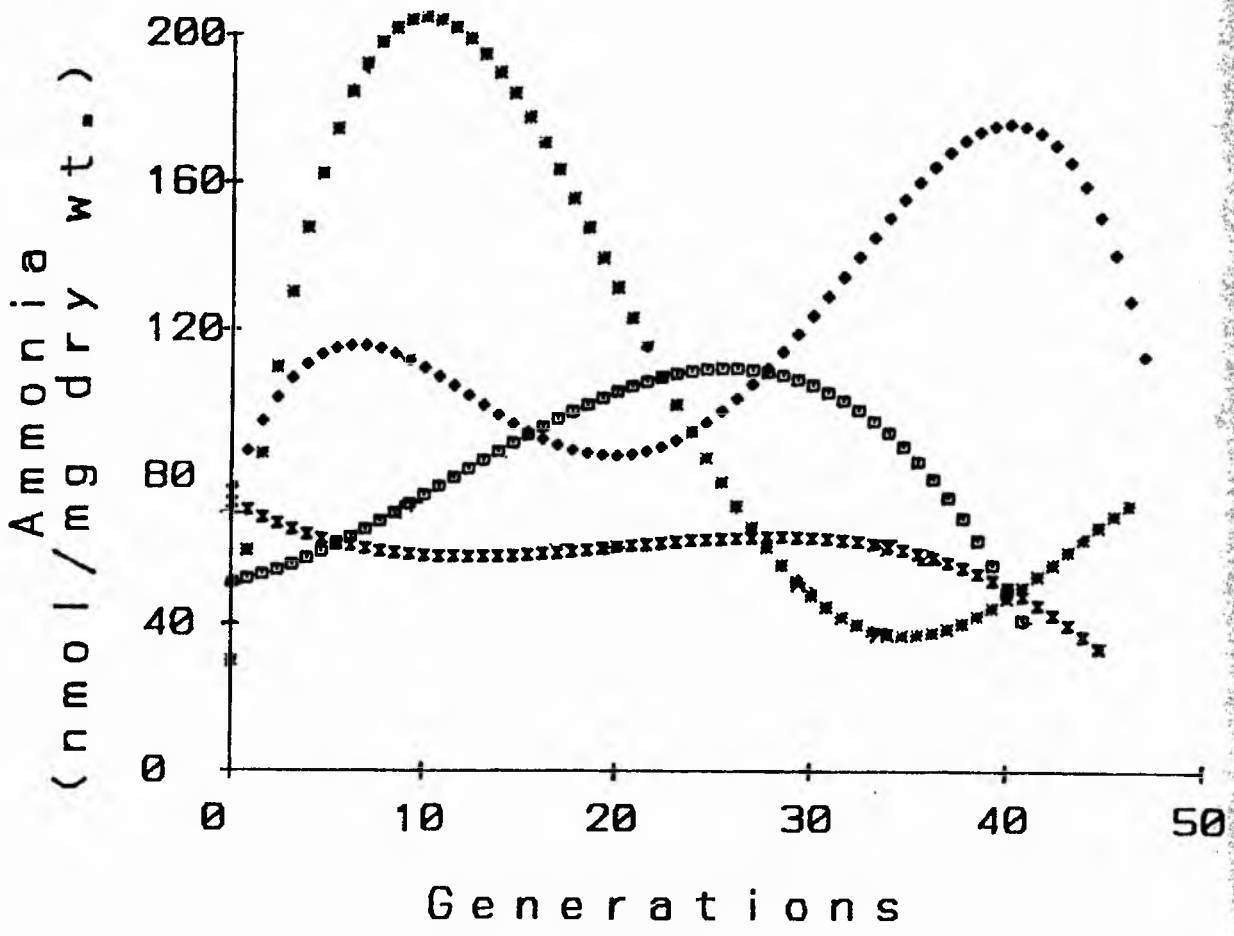


Figure - 6.4.1.3a - Intracellular concentration of ammonia in gdh$^+$ cells growing under carbon-limited continuous culture.

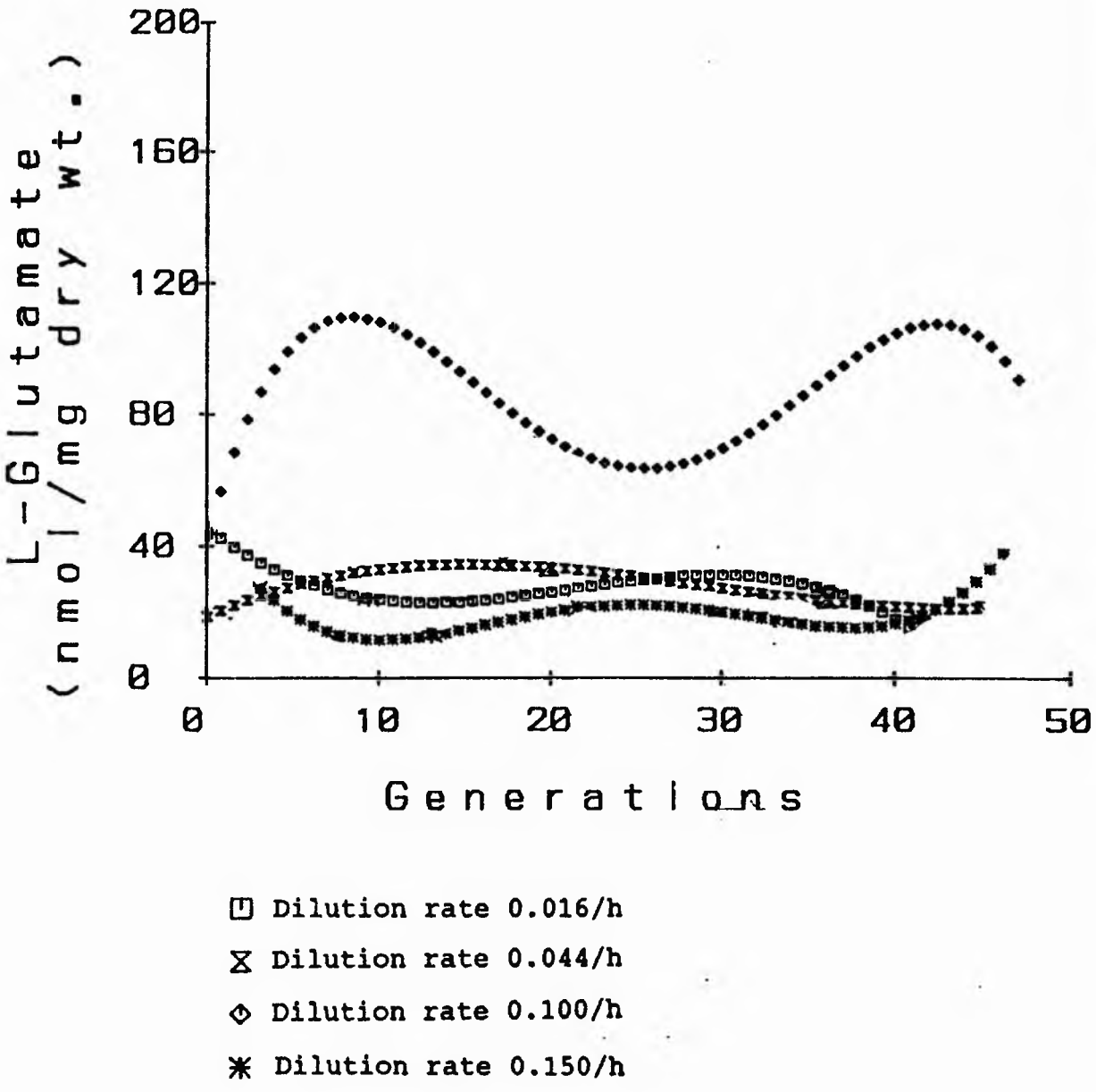


Figure 6.4.1.3b - Intracellular concentration of L-glutamate in gdh^+ cells growing under carbon-limited continuous culture.

6.4.1.4 - Effect of growth rate (carbon-limitation) on plasmid gene expression.

Oscillations in penicillinase activity (figure 6.4.1.4) which are in phase with NADP-GDH activity oscillations, (figure 6.4.1.2a) and also with ammonia and L-glutamate concentrations are probably due to the appearance of a changing mixed population of pCYG4 plasmid positive cells and pCYG4 plasmid negative cells, and may almost certainly be related to competitive advantages and disadvantages conferred on the cells by the pCYG4 plasmid (185).

The NADPH-GDH activity of Saccharomyces cerevisiae increases proportionally with dilution rate (197), (figure 6.4.1.2a) which suggests a regulatory effect stimulated by the need for more enzyme activity by the cells at higher dilution rates. In addition, penicillinase activity, which is proportional to the plasmid copy number, does not change much at low dilution rates (figure 6.4.1.4) in contrast with NADP-GDH activity.

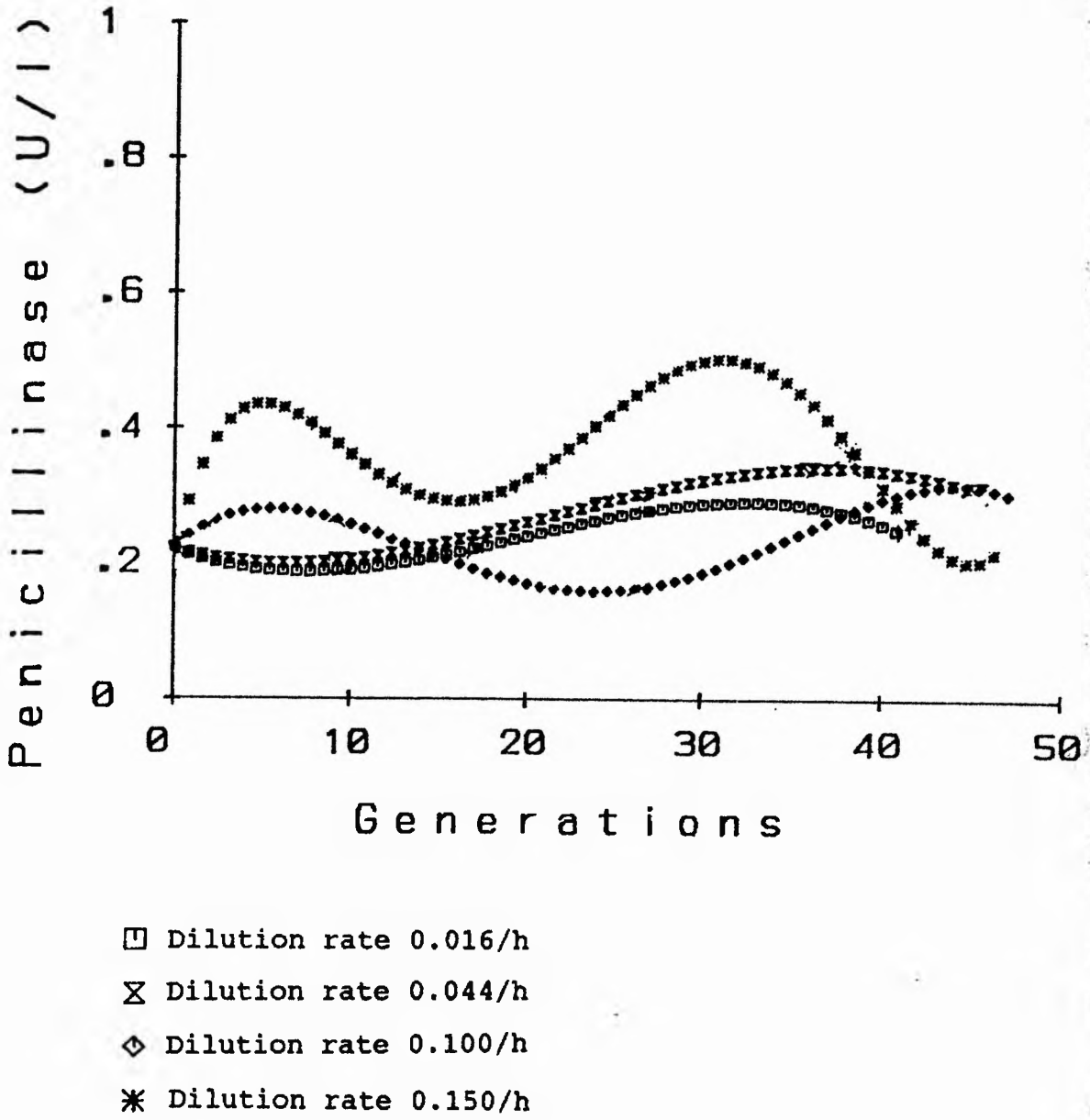


Figure 6.4.1.4 - Penicillinase activity of gdh^+ cells growing under carbon-limited continuous culture.

6.4.2 - Ammonia as limiting substrate.

In batch phase of ammonia limiting substrate experiments, before starting the continuous flow of ammonia limiting substrate medium, the carbon limiting substrate medium (section 3.2.2.2.1) was used as a growth medium, to make sure that all experiments started with the same conditions.

6.4.2.1 - Estimation of maximum specific growth rates of the gdh^+ cells.

Maximum specific growth rates for the gdh^+ cells were determined as indicated in table 6.4.2.1, both in batch mode (at chemostat startup) and, following maintenance of steady state for approximately 45 generations, by washout kinetics (see section 3.2.2.5). There is a significant difference between the values under (a) (under carbon-limiting substrate) and (b) (under ammonia-limiting substrate). One possible explanation is, when the ammonia concentration is very low in the culture medium (e.g. in the steady state under ammonia limitation) the GOGAT pathway becames the main pathway for ammonia assimilation (pathway more energetically expensive than NADP-GDH for the cells). Using the GOGAT pathway cells have to spend more energy (ATP) to assimilate ammonia rather than to spend with the growth. These results are in contrast with carbon limiting substrate experiments in which the NADP-GDH pathway was used as a major pathway. Furthermore, the slow growth rate cannot be related with the

loss of the plasmid, since both penicillinase and NADP-GDH activities are detectable (see figures 6.4.2.4 and 6.4.2.2a). Due to the low growth rate (lower than under carbon-limited substrate experiments) only three experiments were carried out under ammonia limitation) (see table 6.4.2.1).

Table 6.4.2.1. - Maximum specific growth rates of gdh^+, determined during chemostatic experiments (under ammonia limitation), as follows:

(a) During exponential batch growth (chemostat startup).

(b) By washout kinetics from 3 different steady states.

Experiment	(a) μmax (batch h^-)	Steady State growth rate	(b) μmax (washout kinetics)
1	.246	.025	.150
2	.240	.050	.135
3	.255	.088	.165

6.4.2.2 - NADP-GDH and GOGAT activities.

At the three ammonia-limited dilution rates employed GOGAT activity was much greater than in carbon-limiting experiments and was proportional to both dilution rate and NADP-GDH activity (figure 6.4.2.2a).

This high GOGAT activity (figure 6.4.2.2b) , which was proportional to the dilution rate, can be explained by the greater need for more protein at higher dilution rate. Cooper (151) found that, under either carbon or nitrogen-limitation, there is a 10 fold excess of NADP-GDH activity. In contrast with the present results in that while under carbon-limitation, a 10 fold excess of NADP-GDH activity over GOGAT activity was indeed found, under nitrogen-limitation (ammonia as limiting substrate) this difference was reduced (caused by an increase in GOGAT - see figures 6.4.2.2a and 6.4.2.2b).

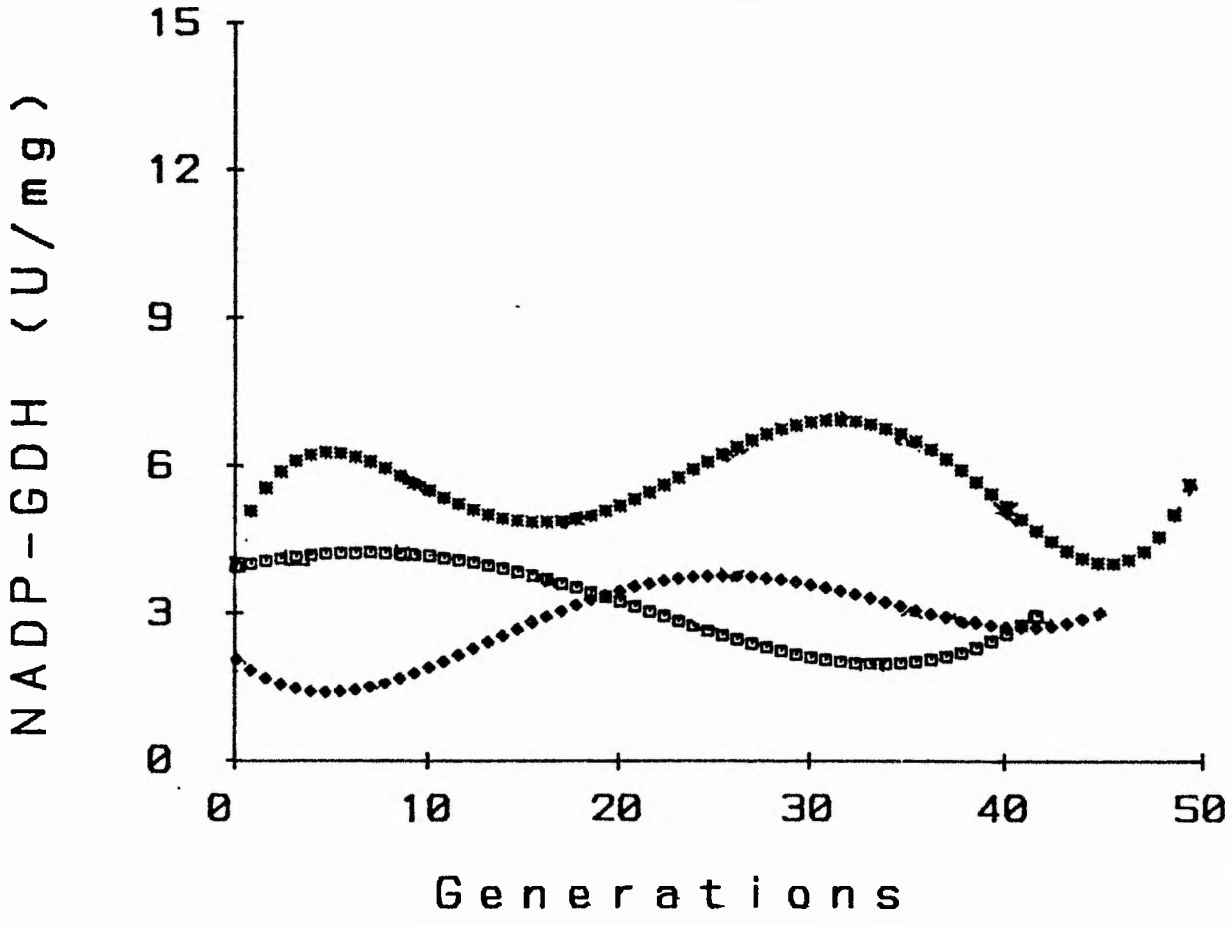


Figure 6.4.2.2a - NADP-GDH activity in gdh^+ cells growing under nitrogen-limited continuous culture.

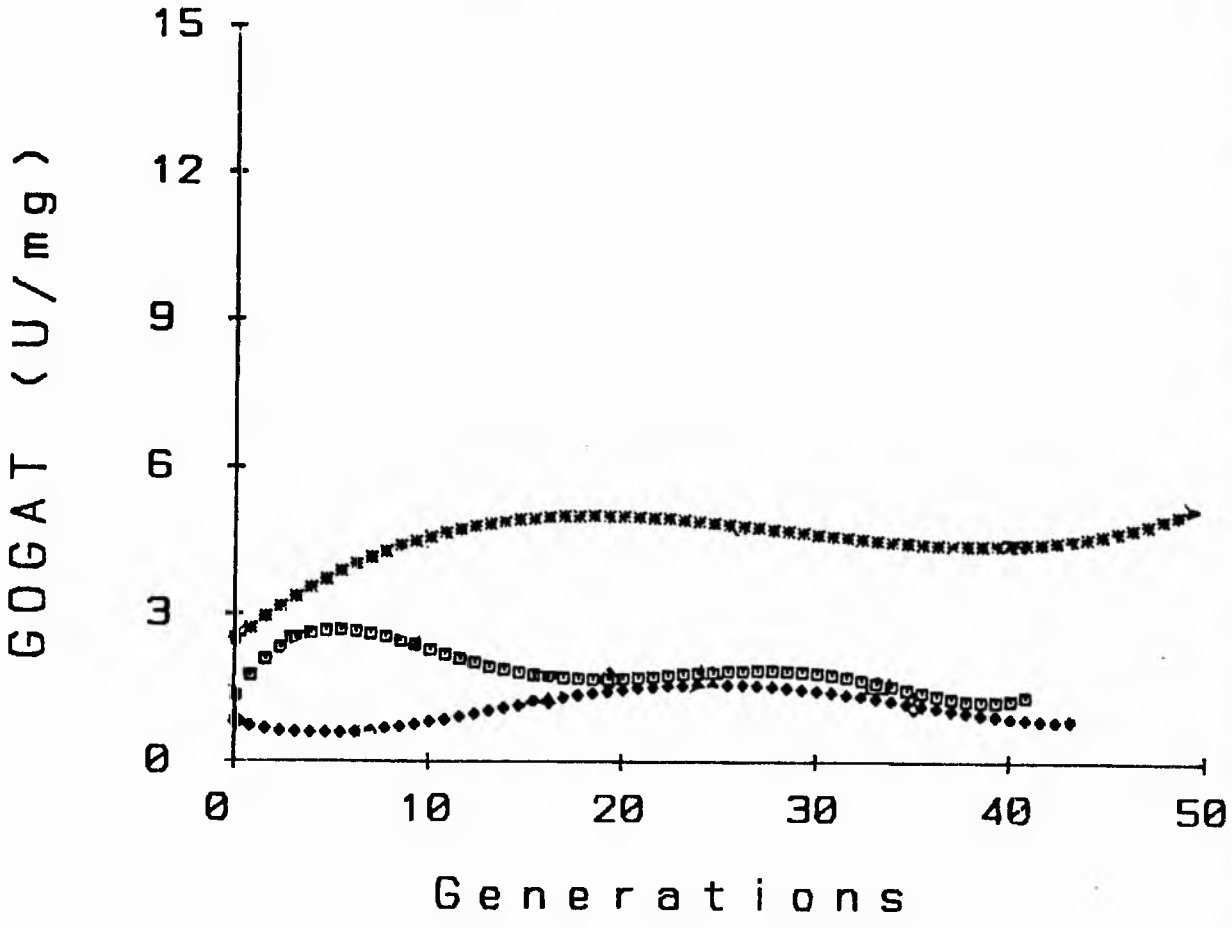


Figure 6.4.2.2b - GOGAT activity in gdh^+ cells growing under nitrogen-limited continuous culture.

6.4.2.3 - Intracellular concentration of ammonia and L-glutamate.

Intracellular ammonia concentrations were found to be inversely proportional to both GOGAT and NADP-GDH activities (figure 6.4.2.3a). This contrasts with L-glutamate intracellular concentrations which were proportional to enzyme activities.

At high growth rate (0.088/h) intracellular ammonia concentration was much lower than at low growth rate (0.025/h). This behaviour was also found under carbon-limitation. However, oscillations with NADP-GDH activity, whose amplitudes increase with intracellular ammonia concentration, may have a greater influence on plasmid instability than ammonia induced repression (see figures 6.4.2.2a and 6.4.2.3a).

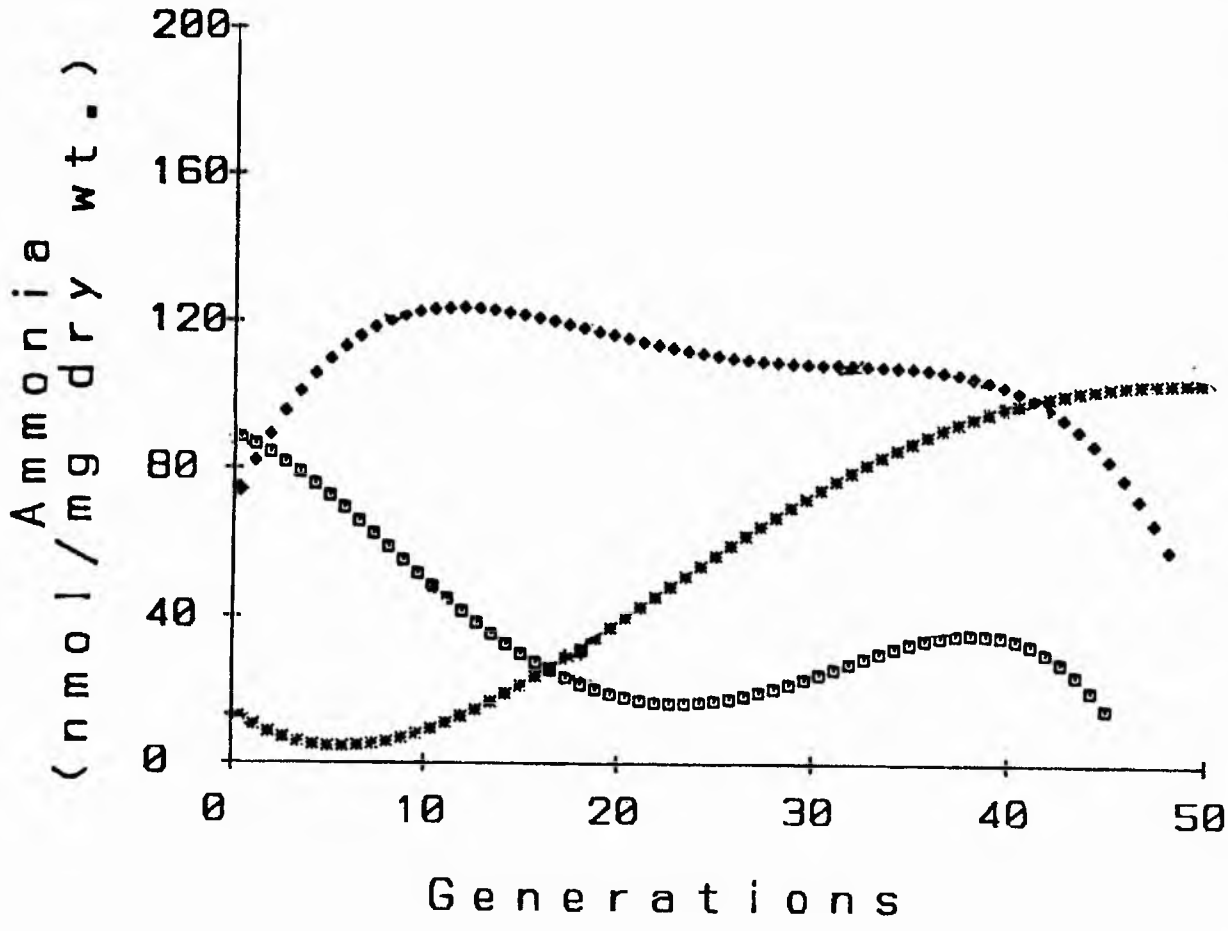


Figure 6.4.2.3a - Intracellular ammonia concentration in gdh$^+$ cells growing under ammonia-limited continuous culture.

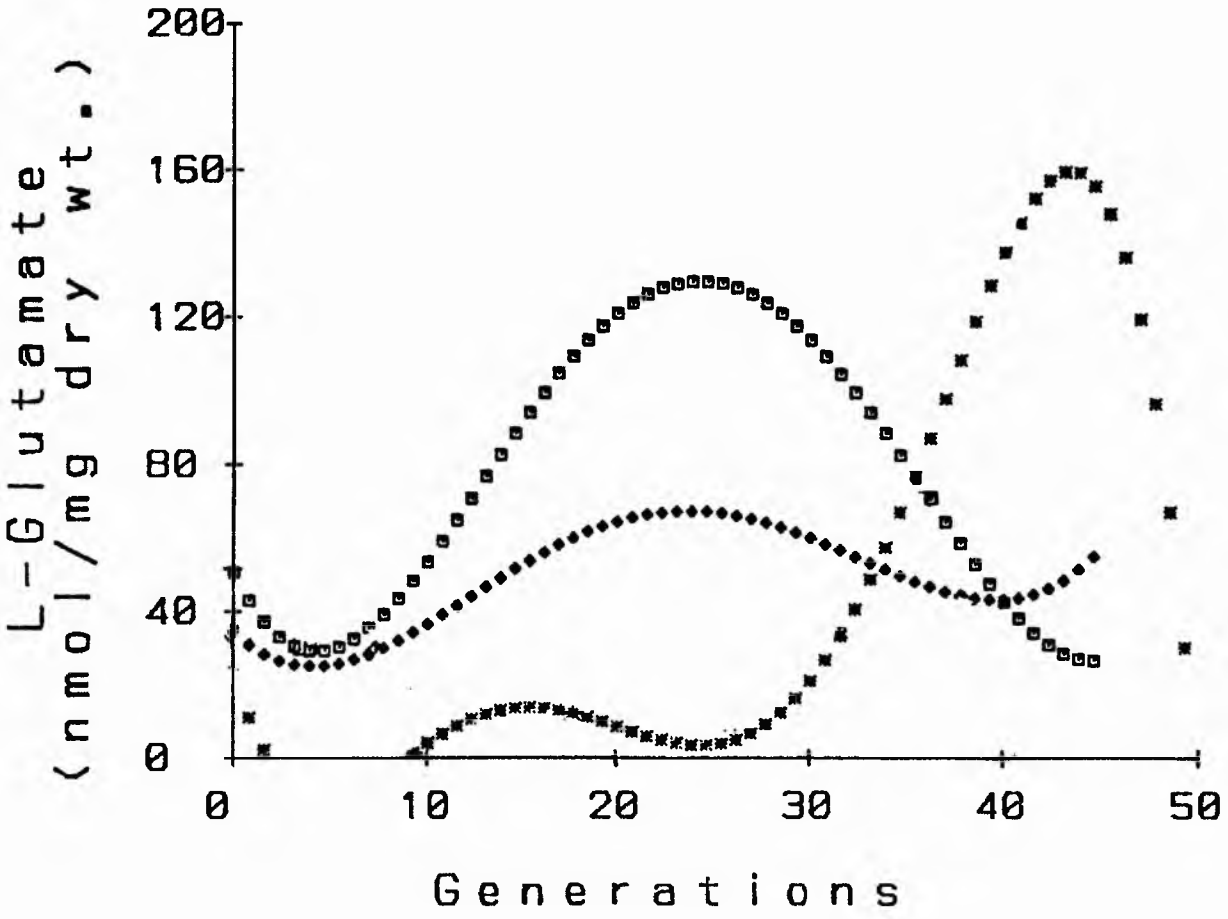


Figure 6.4.2.3b - Intracellular L-glutamate concentration in gdh^+ cells growing under ammonia-limited continuous culture.

6.4.2.4 - Effect of growth rate (ammonia-limitation) on plasmid gene expression.

Oscillations in penicillinase activity (figure 6.4.2.4) were in phase with NADP-GDH activity as found under carbon limitation. However, the frequency of oscillations of penicillinase was independent of growth rate variations, in contrast with experiments under carbon-limitation.

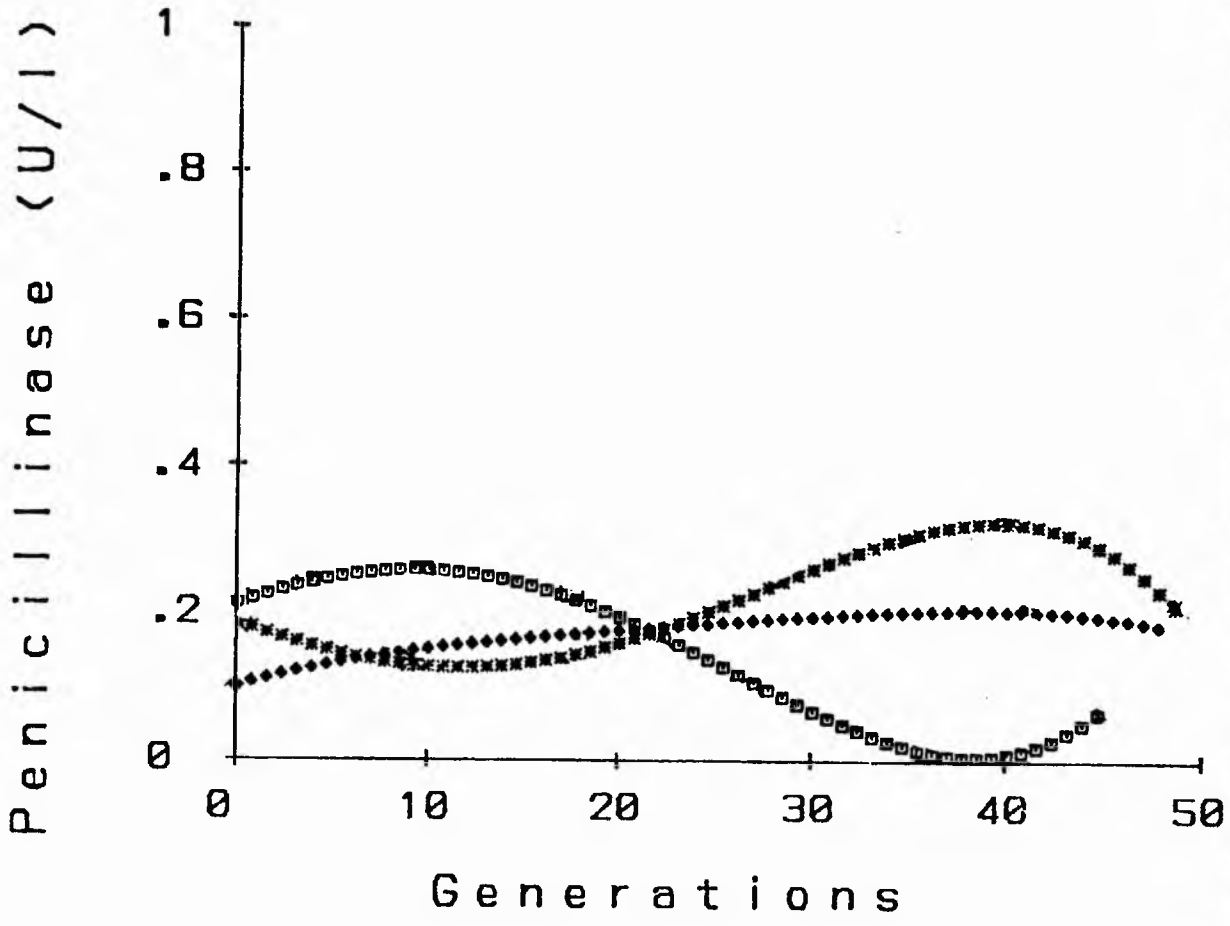


Figure - 6.4.2.4 - Penicillinase activity in gdh$^+$ cells growing under ammonia-limited continuous culture.

6.4.3 - L-Glutamate as limiting substrate.

In the initial batch phase of experiments under L-glutamate-limited continuous culture, carbon limiting substrate (section 3.2.2.2.1) was used as a growth medium, to make sure that all experiments started under the same conditions.

Maximum specific growth rates for gdh^+ cells were determined as indicated in table 6.4.3, both in batch mode (at chemostat startup) and, following maintenance of steady state for approximately 45 generations, by washout kinetics (see section 3.2.2.5). There is a significant difference between the values under (a) (under carbon limitation substrate) and (b) (under L-glutamate-limitation), but no significant difference was observed between (b) (under L-glutamate-limitation) and (b)(under ammonia-limitation). This small specific growth rate under L-glutamate-limitation can be explained by the extremely low NADP-GDH activity (figure 6.4.3a) and an undetectable GOGAT activity. This extremely low NADP-GDH activity and the absence of GOGAT activity can be explained by an L-glutamate regulatory effect on both enzymes. These results are in agreement with Roon and Evan (160), who observed that L-glutamate could act as a repressor of NADP-GDH (150). The present results suggests that this repression is effective even if the NADP-GDH gene is encoded in a plasmid. In addition the GOGAT activity was also repressed.

However, Grenson et. al. (220) pointed out that in

gdh(-) cells, when the nitrogen source was changed to L-glutamate the growth rate increased to over 0.2/h. In contrast, the present results show the maximum specific growth rate to be about 0.15/h, similar to the gdh$^-$ cells. This behaviour can be explained by the need of the presence of the plasmid in the cells for L-leucine production purposes, since the BC55 Leu$^-$ cells cannot grow in the absence of L-leucine. The loss of penicillinase activity and a low level expression of NADP-GDH activity could be due to a low level of plasmid in the cells, or a result of partial loss of these enzyme activities. Kleinman et. al. (229) have suggested that the loss of the phenotype of Saccharomyces cerevisae with plasmid pJDB248 could result from either loss of the whole chimeric plasmid or partial loss of the Leu2 function from the plasmid population.

Table 6.4.3. - Maximum specific growth rates of the gdh($^{+}$) cells, determined during chemostatic experiment (under L-glutamate limitation), as follows:

(a) During exponential batch growth (chemostat startup).

(b) By washout kinetics from 2 different steady states.

Experiment	(a) μmax (batch h^{-})	Steady State growth rate	(b) μmax (washout kinetics)
1	.222	.022	.150
2	.235	.088	.145

Using L-glutamate as nitrogen source only two sets of experiments were carried out using two different dilution rates. In both experiments, the penicillinase and GOGAT activities were undetectable. At a dilution rate of 0.022/h, NADP-GDH activity was extremely low (figure 6.4.3a), with a slightly higher activity for NAD-GDH activity (catabolic enzyme - figure 6.4.3a). At 0.088/h, the NADP-GDH activity was undetectable after 5 generations, in agreement with Gonzalez et. al. (170) and Roon and Even (160), who proposed that L-glutamate could act as a repressor of NADP-GDH.

After 10 generations at a dilution rate of 0.088/h, when NADP-GDH or GOGAT activities were undetectable, the chemostat was switched to carbon-limitation for 30 generations but, even with these conditions, the activities were undetectable (NADP-GDH and GOGAT activities) and the maximum specific growth rate was about 0.120/h.

In both dilution rate experiments (table 6.4.3) the intracellular concentration of ammonia was constant and extremely low (figure 6.4.3b), with a high intracellular concentration of L-glutamate (figure 6.4.3c). The presence of a low concentration of ammonia in these cells (figure 6.4.3b) can be explained by the increase in activity of the catabolic NAD-GDH (figure 6.4.3a) which converts L-glutamate to ammonia and 2-oxoglutarate, in accordance with Roon and Even (159), who pointed out that the specific activity of the NAD-GDH rises rapidly in yeast supplied with L-glutamate as the sole source of nitrogen.

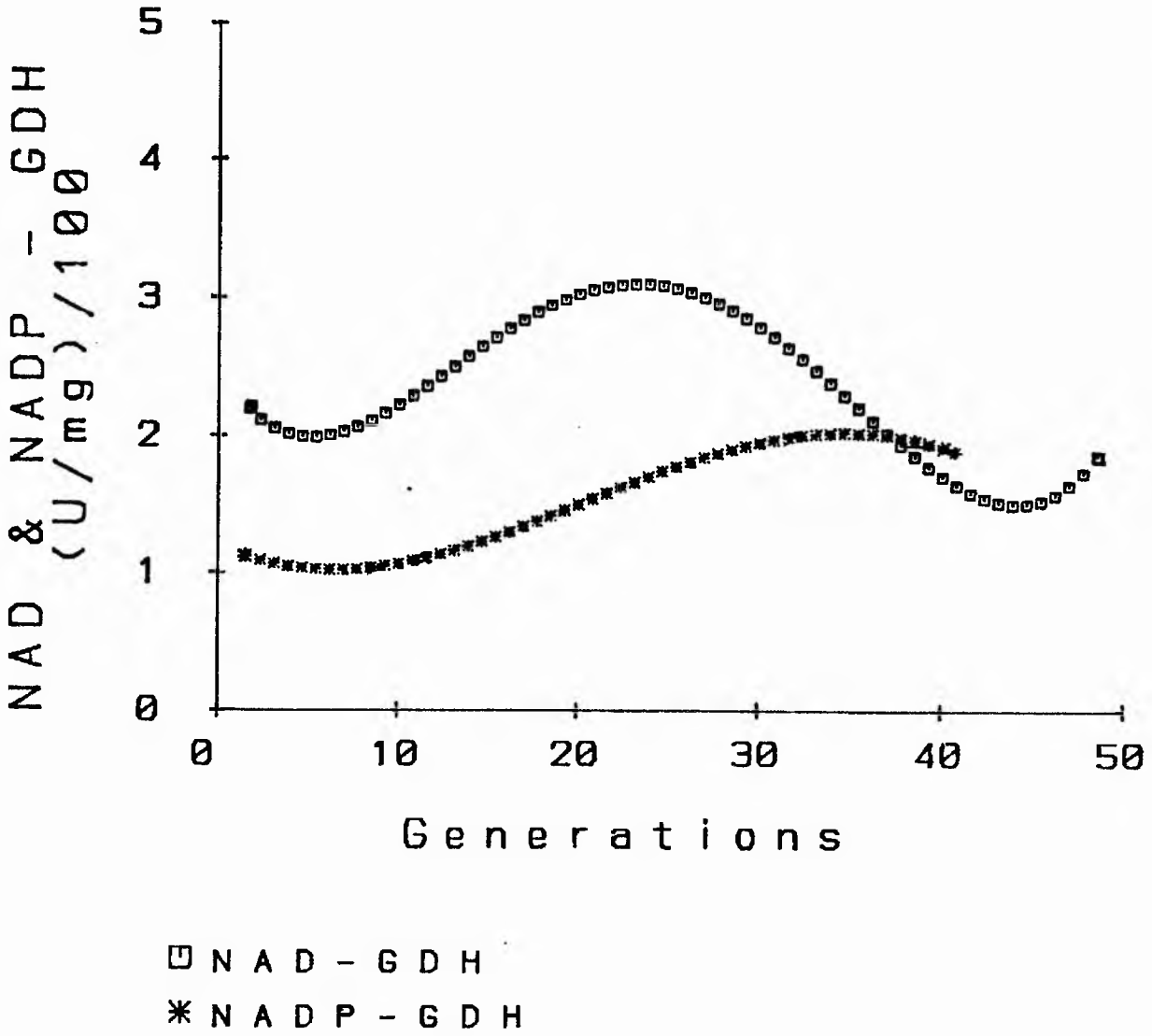
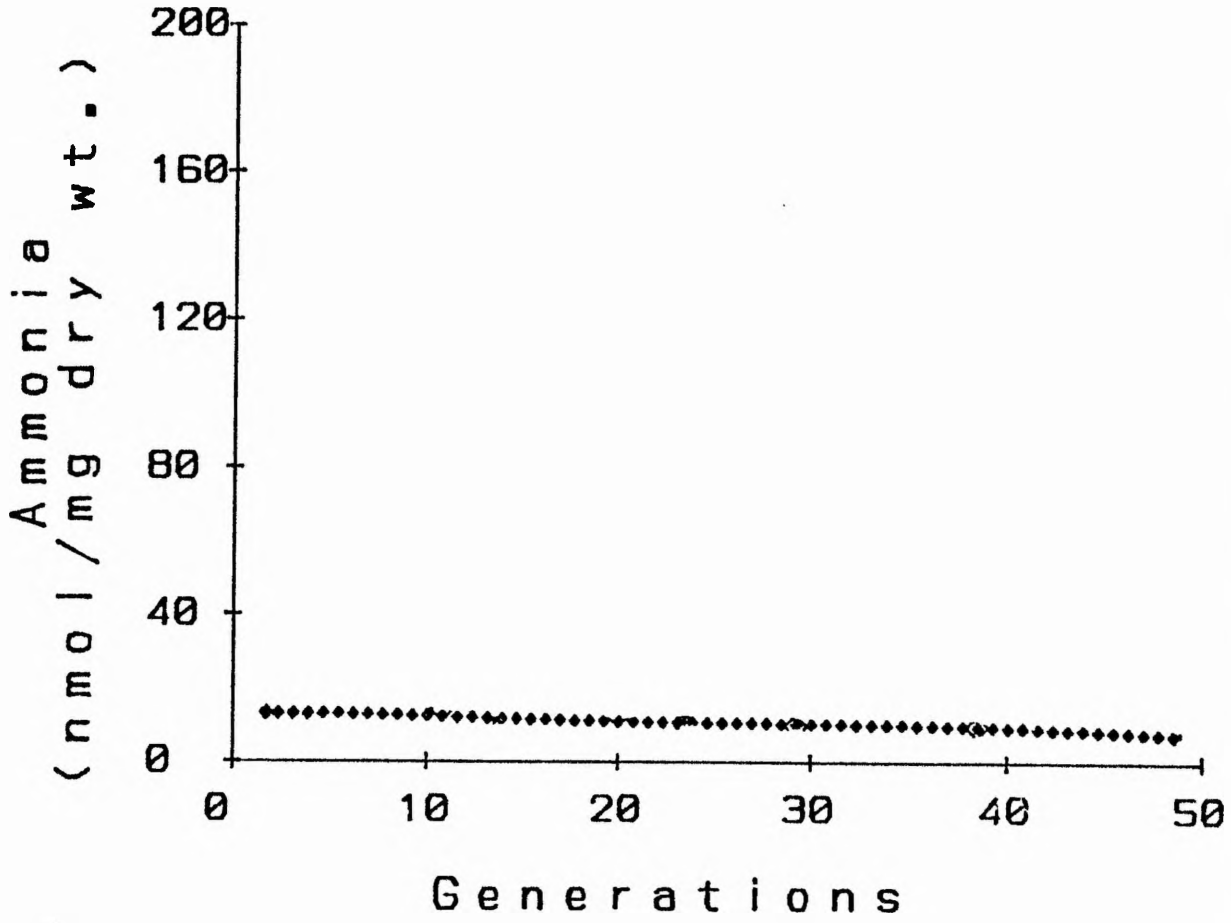
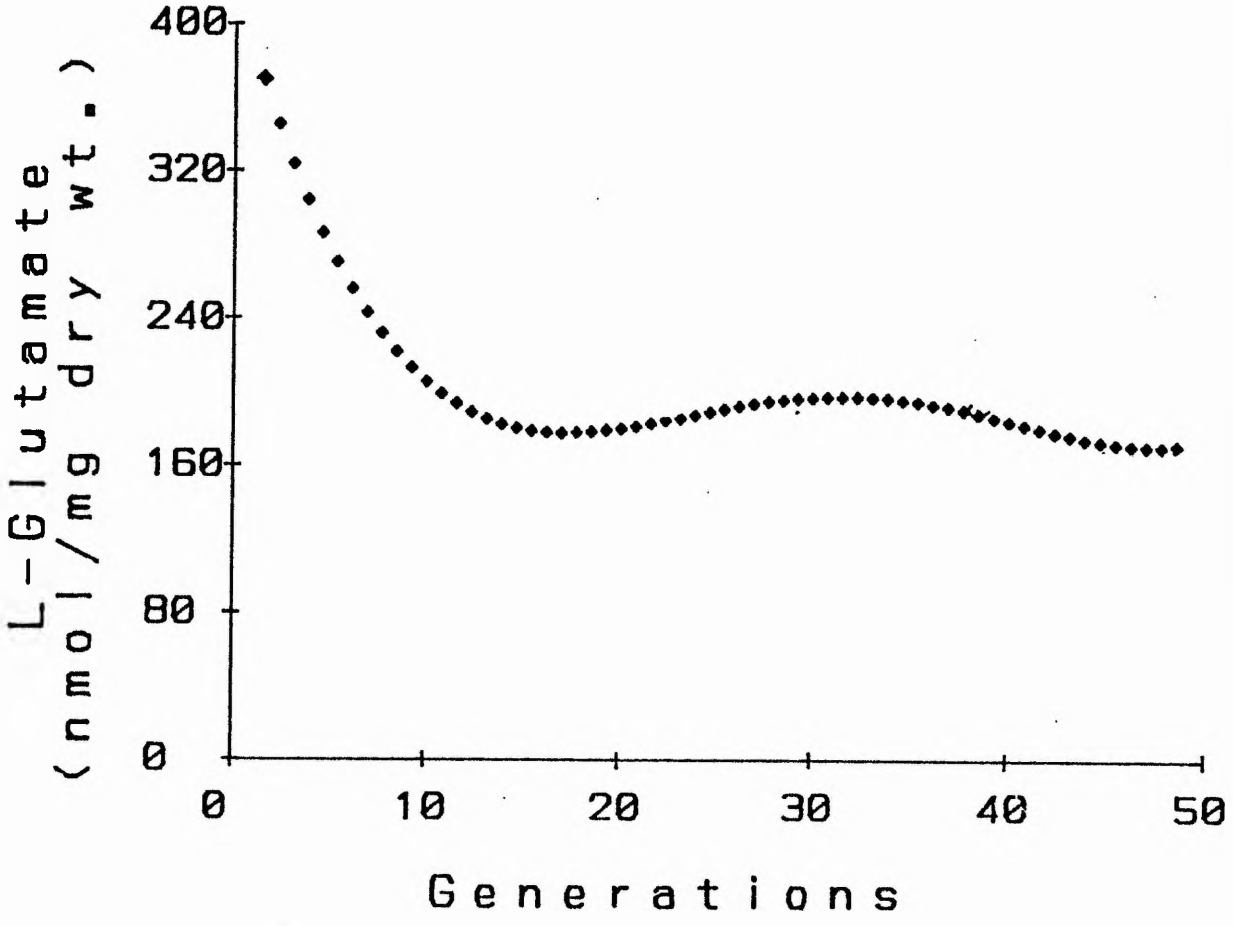


Figure 6.4.3a - NADP-GDH and NAD-GDH activities of gdh^+ cells growing under L-glutamate-limited continuous culture (dilution rate 0.022/h).



◈ Dilution rate 0.022/h

Figure 6.4.3b - Intracellular concentration of ammonia in gdh^+ cells growing under L-glutamate-limited continuous culture (dilution rate 0.022/h).



◈ Dilution rate 0.022/h

Figure 6.4.3c - Intracellular concentration of L-glutamate in gdh$^+$ cells growing under-L-glutamate continuous culture (dilution rate 0.022/h).

6.5 - Plasmid Instability.

Instability here is defined as the ability of transformed cells to lose plasmid (in part or whole).

Oscillations present in the three sets of experiments using different limiting substrates (carbon (glucose), nitrogen (ammonia and L-glutamate)) are probably a consequence of segregational instability of the plasmid pCYG4 in BC55 cells, and this may be related to:

1 - Plasmid phenotype (copy number, relaxed or stringent type of replication control, physiological consequences of gene expression on the plasmid);
2 - Environmental conditions (183, 184, 185).

6.5.1 - Specific growth rate of Saccharomyces cerevisiae BC55 (pCYG4) (gdh^+).

Maximum specific growth rates of cells growing under carbon-limitation were similar to batch culture ones (table 6.4.1.1). This can be explained by the presence of the plasmid which increases the NADP-GDH activity. On the other hand, plasmid instability (oscillations) is proportional to dilution rate. Kleinman et. al. (229), pointed out an increase in instability proportional to growth rate in the chimeric 2μm plasmid. However, Kleinman et. al., (229) also suggested that the probability of plasmid loss increases as

the growth rate decreases, because the lesser amount of material donated by the mother cell into the small bud will reduce the chance of the plasmid being donated. However, Koizumi and Aiba (230), have shown that, depending on the plasmid and the host cells, the plasmid copy number can increase, decrease or stay constant with increase in the dilution rate. The present experiments have been carried out in haploid cells (BC55 pCYG4) and the low growth rate found in these cells using nitrogen as limiting substrate, might differ from diploid cells. Mead et. al. (231) have shown that a recombinant plasmid with a higher copy number in the diploid cells does not reduce the maximum specific growth rate of its host to any significant extent.

The environment can also be an important factor for plasmid instability, for example, at high dilution rate using carbon as limiting growth, the activity of NADP-GDH and penicillinase rises, indicating a increase of plasmid copy number in the cells, however as a product and a pressure factor L-leucine will be increased, too. Now cells without plasmid can increase in the medium. Firstly, because of segregational instability, which increases in proportion to the dilution rate further increasing the plasmid free cell population and secondly because cells without plasmid grow faster than cells with plasmid, the population of cells without plasmid increases. But, as a consequence of a decrease in cells with plasmid the concentration of L-leucine decreases. Cells without plasmid cannot grow in the absence of L-leucine, therefore the concentration of cells with plasmid will tend to increase again.

Under nitrogen limitation, instability (oscillations) is similar to the carbon limiting experiments. The difference is that now the concentration of ammonia in the medium is low, consequently the GOGAT pathway will be activated, even in the cells without plasmid. The low maximum specific growth rate (smaller than under carbon-limitation) could be explained by the increase in activity of the GOGAT pathway.

With L-glutamate as nitrogen limiting substrate the NADP-GDH and GOGAT activities were undetectable and the maximum specific growth rate was the same as the BC55 with the plasmid YEp13. However these cells (leu(-)) need L-leucine to grow, consequently the presence of the plasmid is essential, conferring pratically the same growth characteristics as the BC55 YEp13.

6.5.2 - Plasmid phenotype.

Futcher and Cox (232) pointed out that the amount of plasmid per cell can be random distributed using a simple model for a plasmid system. The model assumes that the copy number is maintained at some number n in each cell, that the number of plasmids doubles to 2n before cell division, and that these 2n copies are randomly distributed between the daughter cells. If instability were due mainly to random plasmid partition, then instability would satisfy the

equation $i=2^{-2n}$, but plasmid instabilities are far higher than predicted. This is true for the 2μm itself, as well as for chimeric plasmids based upon it, which confer a disadvantage for the cells with high plasmid copy number.

Petes and Williamson (233), have pointed out that, under normal growth conditions, replication of 2μm circle strictly conforms to the cell cycle regulation. However, Jayaram et. al., (234) have shown that plasmid population can, in certain circunstances, double more frequently than each cell generation and that the plasmid can replicate more frequently that DNA. Furthermore, they have also suggested that the plasmid 2μm can increase its copy number following cytoduction or germination. Consequently, the synthesis of plasmids and products increases, including the complementing product which can decrease the growth of plasmid-containing cells to some degree (186), which can reduce the host's growth rate by 28% (187). This selective disadvantage conferred by the plasmid would be proportional to their copy number (182).

Oscillations in NADP-GDH and penicillinase activities can be explained by:

1 - Increase of the concentration of cells with plasmid;

2 - Increase of plasmid copy number in the cells which reduces the growth rate conferring advantages to cells without plasmid. Furthermore, with an increase of dilution rate the activity of the enzymes rises, and, consequently,

the concentration of products. As a product L-leucine concentration would increase as well, then cells without plasmid (Leu$^-$, which cannot grow in a lack of L-leucine) can utilise the L-leucine present in the medium, and because there is L-leucine in the medium and these cells do not have the plasmid, they can grow faster the cells with plasmid, consequently their population increases. But, because their population rises and the population of plasmid-containing cells decreases, the concentration of L-leucine will decrease. Then plasmid-free cells decline again and the plasmid-containing cells once more increase their numbers and the oscillatory cycle continues.

7. SUMMARY.

A laboratory scale fermenter control system, based on a Z80 microprocessor operating at 4 MHz, has been developed. The system has the following features: 1 -» 8 Kbytes of RAM and 4 Kbytes of EPROM (with extra space for another 4 Kbytes); 2 -» Two serial links (RS232c); 3 -» An analog/digital converter with an 8 channel multiplexer and an opto-triac system for switching control devices (heater, peristaltic pumps); 4 -» Four interface cards to connect the parameter sensors.

This system has the advantage that it is based on well established (though not state-of-the-art) microelectronic technology (Z80 chips) widely available in the world at reasonable prices. This, plus the fact that both hardware and software were locally designed, means that it is truely 'portable' in the sense that, with limited resources, copies can easily be constructed in devoloping countries (the idea is that Mark-II will be built in North East Brasil to support research into yeast fermentation technology).

The control system was evaluated in batch and chemostatic growth modes. Both modes were used to investigate the growth kinetics of a genetically engineered Saccharomyces cerevisiae strain (BC55) carrying the plasmid pCYG4 which directs 10 fold more NADP-GDH activity than wild type cells.

Batch culture experiments showed that the presence of

plasmid increased ammonia uptake by the cells but did not improve biomass or ethanol yield compared to wild type cells.

Under carbon-limitation NADP-GDH activity was in phase with GOGAT and penicillinase activity. NADP-GDH and GOGAT activities were inversely proportional to intracellular ammonia concentration and proportional to intracellular L-glutamate concentration.

Under nitrogen limitation (using ammonia as limiting substrate) GOGAT activity increased, but NADP-GDH and penicillinase activity remained at the same level as under carbon limitation. Using L-glutamate as nitrogen source NADP-GDH activity was very low and GOGAT and penicillinase activities were undetectables with an increase in NAD-GDH activity.

Oscillations found in enzyme activities and intracellular metabolite concentration under carbon and nitrogen limitation experiments are a consequence of a mixed culture (the presence of cells with plasmid GDH (gdh^+ cells) and cells without plasmid GDH (gdh^- cells)) with changing concentrations of the 2 populations.

```
;-------------------------------------------------------------;
;                  PCS - Processor Control System             ;
;               PCS-BIOS (Basic Input/Output System)          ;
;                    University of St. Andrews                ;
;            Department of Biochemistry and Microbiology      ;
;                         Master Version 2.0                  ;
;                      By Jose Luiz de Lima Filho             ;
;                              Oct. 1984                      ;
;-------------------------------------------------------------;

;----------------------;
;       EQUATES        ;
;----------------------;

        ORG     0000H           ;INITIAL ADDRESS (ROM)
TTY     EQU     010H            ;FIRST SERIAL PORT (USART1) ADDRESS
CTTY    EQU     011H            ;STATUS OF USART1
AD      EQU     030H            ;ANALOG/DIGITAL CONVERTER ADDRESS
CLK     EQU     040H            ;REAL TIME CLOCK ADDRESS
TTY1    EQU     050H            ;SECOND SERIAL PORT (USART2) ADDRESS
CTTY1   EQU     051H            ;STAUTS OF USART2
DA      EQU     080H            ;SWITCH ON/OFF ADDRESS
PRINT   EQU     0DCFFH
        LD      SP,0DDFFH
        LD      HL,0DCFFH

;----------------------------;
;    INITIALIZE USARTs       ;
;----------------------------;

        LD      A,07AH          ;MODE OPERATION (USART1)
        OUT     (CTTY),A
        LD      A,035H          ;MODE INSTRUCTION (USART1)
        OUT     (CTTY),A
        LD      A,07AH          ;MODE OPERATION (USART2)
        OUT     (CTTY1),A
        LD      A,035H          ;MODE INSTRUCTION (USART2)
        OUT     (CTTY1),A
        LD      A,01H
        LD      (PRINT),A

;-----------------------------------;
;   INITIALIZE KEYBOARD FUNCTIONs   ;
;-----------------------------------;

INICIO: LD      C,0AH
        CALL    STOUT
        LD      C,00DH
        CALL    STOUT
        LD      C,'>'           ;CURSOR CODE
        CALL    STOUT           ;SEND A '>' TO THE VIDEO
        CALL    STIN            ;GET A CODE
        LD      A,C
        CP      'K'
        JP      Z,KMASTER       ;IF K GO TO KMASTER
        LD      A,C
        CP      'V'
        JP      Z,TRSRRA        ;IF V GO TO TRSRRA
        LD      A,C
        CP      'F'
```

```
        JP      Z,LOOPC         ;IF F GO TO LOOPC
        LD      A,C
        CP      'A'
        JP      Z,CMDA          ;IF Z GO TO CMDA
        LD      A,C
        CP      'C'
        JP      Z,CMDC          ;IF C GO TO CMDC
        LD      A,C
        CP      'D'
        JP      Z,CMDD          ;IF D GO TO CMDD
        LD      A,C
        CP      'E'
        JP      Z,CMDE          ;IF E GO TO CMDE
        LD      A,C
        CP      'L'
        JP      Z,CMDL          ;IF L GO TO CMDL
        LD      A,C
        CP      'I'
        JP      Z,INICIO        ;IF I GO TO INICIO
        LD      A,C
        CP      0DH
        JP      Z,INICIO        IF CR GO TO INICIO

;-------------------------;
;       ERRO CHECK        ;
;-------------------------;

ERRO:   CALL    STOUT
        LD      C,'?'
        CALL    STOUT
        JP      INICIO

;-------------------------;
;     OUPUT OF DATA       ;
;-------------------------;

STOUT:  LD      A,(PRINT)
        XOR     01H
        JP      Z,TERM
PRT1:   IN      A,(CTTY1)
        AND     02H
        JP      Z,PRT1
        IN      A,(TTY1)
        CP      02H
        JP      NZ,PRT1
PRT:    IN      A,(CTTY1)
        AND     01H
        JP      Z,PRT
        LD      A,C
        OUT     (TTY1),A
        CALL    TERM
        RET
TERM:   IN      A,(CTTY)
        AND     01H
        JP      Z,STOUT
        LD      A,C
        OUT     (TTY),A
        RET
```

```
;-------------------------------------;
;  CONVERTER ASCII TO HEXADECIMAL     ;
;-------------------------------------;


;CONVERSION ASC -> HEX
ASCHEX  LD      A,C
        CP      040H
        JP      M,NUM
        LD      A,C
        AND     0FH
        ADD     A,09H
        JP      ENTRE
NUM:    LD      A,C
        AND     OFH
ENTRE:  RET

;-------------------------------------;
;      DISPLAY DATA ON THE VIDEO      ;
;      BETWEEN TWO ADDRESSES          ;
;-------------------------------------;

CMDD:   CALL    CARHL
        LD      C,','
        CALL    STOUT
        CALL    LBYTE
        LD      D,A
        CALL    LBYTE
        LD      E,A
        CALL    STIN
        CP      0DH
        JP      NZ,ERRO
NOVO    LD      B,010H
        CALL    CRLF
        CALL    ECOHL
VELHO   LD      C,' '
        CALL    STOUT
        LD      C,(HL)
        CALL    ECO1
        LD      A,D
        XOR     H
        JP      NZ,DEC1
        LD      A,E
        XOR     L
        JP      NZ,DEC1
        JP      INICIO
DEC1:   INC     HL
        DEC     B
        JP      Z,NOVO1
        JP      VELHO
```

```
;-------------------------;
;      INPUT OF DATA      ;
;-------------------------;

STIN:   IN      A,(CTTY)        ;CHECK STATUS
        AND     02H
        JP      Z,STIN
        IN      A,(TTY)         ;INPUT FROM USART1
        LD      C,A
        CALL    STOUT
        RET
STIN1:  IN      A,(CTTY)
        AND     02H
        RET     Z
        IN      A,(TTY)
        LD      C,A
        CALL    STOUT
        RET

;-----------------------------------------;
;       READ DATA AND CONVERT THEM        ;
;       FROM ASCII TO HEXADECIMAL         ;
;-----------------------------------------;

;BYTE
LBYTE:  CALL    STIN            ;GET DATA
LBYTE   LD      A,C
        CP      030H            ;IF SMALLER THAN 0 ERRO
        JP      M,ERRO
        LD      A,C
        CP      047H            ;IF BIGGER THAN F ERRO
        JP      P,ERRO
        CALL    ASCHEX          ;STORE DATA
        PUSH    BC
        RLCA
        RLCA
        RLCA
        RLCA
        LD      B,A             ;SEND THE LSB (BITS 4,5,6 AND 7)
        CALL    STIN
        LD      A,C
        CP      030H
        JP      M,ERRO
        LD      A,C
        CP      047H
        JP      P,ERRO
        CALL    ASCHEX
        ADD     A,B
        POP     BC
        RET
```

```
        CP      020H
        JP      Z,ESPAC
        LD      A,C
        CP      0DH
        JP      Z,INICIO
        CALL    LEBY
        LD      (HL),A
        LD      C,' '
        CALL    STOUT
ESPAC:  INC     HL
        DEC     B
        JP      NZ,NOVLIN
        JP      NOVME

;-------------------------------------------------------;
;   STATUS TERMINAL, INPUT AND ECHO OF THE SYSTEM       ;
;         CONVERSION HEXADECIMAL -> ASCII               ;
;-------------------------------------------------------;

ECO1:   PUSH    DE
        PUSH    BC
HEXASC: LD      A,C
        LD      E,C
        LD      D,02H
        AND     0F0H
        RRCA
        RRCA
        RRCA
        RRCA
        LD      B,A
DOIS:   CP      0AH
        LD      A,B
        JP      M,NUMBER
        LD      A,B
        ADD     A,037H
        JP      DUPLA
NUMBER: ADD     A,030H
DUPLA:  LD      C,A
        CALL    STOUT
        DEC     D
        JP      Z,FIM
        LD      A,E
        AND     0FH
        LD      B,A
        JP      DOIS
FIM:    POP     BC
        POP     DE
        RET

;ECHO HL REGISTERS CONTENTS

ECOHL:  LD      C,H
        CALL    ECO1
        LD      C,L
        CALL    ECO1
        RET


;LOAD  HL  REGISTERS
```

```
;-----------------------------------------------;
;  LOAD A SPECIFIC AREA OF THE RAM MEMORY       ;
;  WITH DATA FROM THE KEYBOARD                  ;
;-----------------------------------------------;

CMDC:   CALL    CARHL
        CALL    STIN
        CP      0DH
        JP      NZ,ERRO
VOLTB:  LD      B,010H
        CALL    CRLF
        CALL    ECOHL
VOLTC:  LD      C,' '
        CALL    STOUT
        CALL    STIN
        CP      0DH
        JP      Z,FIMC
        CALL    LEBY
        LD      (HL),A
        INC     HL
        DEC     B
        JP      Z,VOLTB
        JP      VOLTC
FIMC:   LD      M,0C9H
        JP      INICIO

;-----------------------------------------------------;
;       EXECUTE A SEQUENCIAL DATA OF INFORMATION      ;
;       BEGIN FROM A SPECIFIC ADDRESS                 ;
;-----------------------------------------------------;


CMDE:   CALL    CARHL
        CALL    STIN
        CP      0DH
        JNZ     INICIO
        PUSH    AF
        LD      DE,SALSP
        PUSH    DE
        JP      M
SALSP:  POP     AF
        JP      INICIO

;-------------------------------------------------------;
;     ALTER THE CONTENTS OF A SPECIFIC RAM ADDRESS      ;
;-------------------------------------------------------;


CMDA:   CALL    CARHL
        CALL    STIN
        CP      0DH
        JP      NZ,ERRO
NOVLIN: LD      B,010H
        CALL    CRLF
        CALL    ECOHL
        LD      C,' '
        CALL    STOUT
NOVME:  LD      C,(HL)
        CALL    ECO1
        CALL    STIN
```

```
CARHL:  LD      C,' '
        CALL    STOUT
        CALL    LBYTE
        LD      H,A
        CALL    LBYTE
        LD      L,A
        RET

;SEND A CR AND LF TO THE VIDEO DISPLAY


CRLF:   LD      C,0AH
        CALL    STOUT
        LD      C,0DH
        CALL    STOUT
        RET

;-------------------------------------------------;
;   LIMIT OF SAMPLE IN THE DISPLAY BY LINES       ;
;-------------------------------------------------;

NOVO1:  CALL    STIN1
        CP      013H
        JP      Z,CTRLQ
        LD      A,C
        CP      03H
        JP      Z,INICIO
        JP      CTRLQ
CTRLC:  PUSH    AF
        IN      A,(CTTY)
        AND     02H
        JP      Z,RETO1
        IN      A,(TTY)
        CP      03H
        JNZ     RETO1
        POP     AF
        INC     SP
        INC     SP
        JP      INICIO
RETO1:  POP     AF
        EI
        RET

;-------------------------------------------------;
; READ THE CONTENT OF THE RAM OF A SPECIFIC       ;
;                 PARAMETER                       ;
;-------------------------------------------------;

CMDL:   LD      C,0CH
        CALL    STOUT
        LD      C,'V'
        CALL    STOUT
        LD      C,'='
        CALL    STOUT
        CALL    STIN
        LD      A,'P'
        CP      C
        JP      Z,PHL           ;pH ELECTRODE DATA
        LD      A,'O'
        CP      C
```

```
        JP      Z,OXL           ;OXYGEN ELECTRODE DATA
        LD      A,'T'
        CP      C
        JP      Z,TEML          ;TEMPERATURE ELECTRODE DATA
        LD      A,'B'
        CP      C
        JP      Z,BIOL          ;BIOMASS ELECTRODE DATA
        LD      A,'M'
        CP      C
        JP      Z,MOTL          ;ALKALI SYSTEM DATA
        JP      INICIO
PHL:    LD      DE,(0DE22H)     ;pH DATA ADDRESS
        LD      HL,0C000H
        JP      LFIM
OXL:    LD      DE,(0DE24H)     ;OXYGEN DATA ADDRESS
        LD      HL,0C400H
        JP      LFIM
TEML:   LD      DE,(0DE26H)     ;TEMPERATURE DATA ADDRESS
        LD      HL,0C800H
        JP      LFIM
BIOL:   LD      DE,(0DE2AH)     ;BIOMASS DATA ADDRESS
        LD      HL,0D000H
        JP      LFIM
MOTL:   LD      DE,(ODE2AH)     ;ALKALI SYSTEM ADDRESS
        LD      LH,0CC00H
LFIM:   LD      A,00H
        LD      (0DCFFH),A
        CALL    NOVO
        LD      A,01H
        CALL    STOUT
        RET

;-------------------------------------------------;
;       INITIALIZE THE REAL TIME CLOCK AND        ;
;       THE LOOP MASTER SYSTEM                    ;
;-------------------------------------------------;

KMASTER:LD      A,00H           ;STOPS THE RTC
        OUT     (4EH),A

;CLEAR ALL REGISTERS OF THE RTC

        LD      A,00H
        OUT     (44H),A
        OUT     (45H),A
        OUT     (46H),A
        OUT     (47H),A
        OUT     (48H),A
        OUT     (49H),A
        OUT     (4AH),A
        OUT     (4BH),A
        OUT     (4CH),A
        OUT     (4DH),A
        LDT     A,0FFH
        OUT     (4EH),A         ;RE-INITIALIZES THE RTC
        NOP
LOOPC:  IN      A,(045H)        ;ACTIVATES TENS OF MINS. REGISTER
        RRCA
        RRCA
        RRCA
```

```
        RRCA
        AND     03FH
        CP      03FH
        JP      Z,LOOPC
        LD      (0DE04H),A
        CALL    CONT1           ;CALL THE CONTROL SUBROUTINE
        CALL    STIN1
        CP      03H
        JP      Z,CROFF
        LD      A,(0DE04H)
        AND     01H
        LD      HL,0DE00H
        CP      (HL)
        JP      Z,LOOP2
        JP      LOOPC

;CHECK THE RTC FUNCTION

LOOP2:  LD      A,(0DE00H)
        XOR     01H
        LD      (0DE00H),A
        LD      HL,0DE01H
        LD      A,(0DE02H)
        CP      (HL)
        JP      Z,LOOP3
        INC     (HL)
        JP      LOOPC
LOOP3:  CALL    GETP1           ;CALL THE GET SUBROUTINE
        LD      A,00H
        LD      (0DE01H),A
        JP      LOOPC
CORCT:  LD      DE,(0DE32H)
        LD      A,(DE)
        LD      C,A
        LD      A,(0DE44H)
        SUB     C
        RET

;-----------------------------------;
;       CONTROL SUBROUTINE          ;
;-----------------------------------;


CONTROL:LD      C,35            ;OXYGEN CONDITION
        CALL    GEPPO
        LD      A,(0DE1CH)
        CP      E
        JP      M,OX1
        LD      A,0FFH
        SUB     10H
        OUT     (080H),A
        LD      (0DFFCH),A
TEMP:   LD      C,30H           ;TEMPERATURE CONDITION
        CALL    GEPPO
        LD      A,(0DE1DH)
        CP      E
        JP      M,PH
        LD      A,(0DFFCH)
        SUB     080H
        OUT     (080),A
```

```
        LD      (0DFFCH),A
PH:     LD      C,32            ;pH ALKALI CONDITION
        CALL    GEPPO
        LD      A,(0DE1EH)
        CP      E
        JP      M,PH1
        LD      A,(0DFFCH)
        SUB     080H
        OUT     (080H),A
        LD      (0DFFCH),A
        JP      ANTF
PH1:    LD      C,32            ;pH ACID CONDITION
        CALL    GEPPO
        LD      A,(0DE1EH)
        CP      E
        JP      P,ANTF
        LD      A,(0DFFCH)
        SUB     01H
        OUT     (080H),A
        LD      (0DFFCH),A
ANTF:   LD      C,34            ;FOAM CONDITION
        CALL    GEPPO
        LD      A,(0DE20H)
        CP      E
        JP      P,RET1
        LD      A,(0DFFCH)
        SUB     040H
        OUT     (080H),A
        LD      (0DFFCH),A
RET1:   RET

;-------------------------------------------------------;
;    GET DATA FROM THE ANALOG/DIGITAL CONVERTER         ;
;    FROM EACH PARAMETER                                ;
;-------------------------------------------------------;

GEPPO:  LD      HL,0DC00H
        LD      B,064H          ;TAKES 64 POINTS FROM A PARAMETER
        LD      A,00H
GEP1:   CALL    INTF            ;STORE THEM
        LD      (HL),A
        INC     HL
        LD      (HL),00000H
        INC     HL
        DEC     B
        JP      Z,ADD1
        JP      GEP1

;ADD ALL 64 POINTS

ADD1:   LD      (0DFFAH),SP
        LD      B,64H
        LD      SP,0DC00H
        LD      HL,00000H
AD1:    POP     DE
        ADD     HL,DE
        DEC     B
        JP      Z,RAD
        JP      AD1
RAD:    EX      DE,HL
```

```
        LD      (0DFF8H),DE
        LD      SP,(0DFFAH)

;FIND THE AVERAGE VALUE OF 64 POINTS

DIVS:   LD      DE,(0DEF8H)
        EX      DE,HL
        LD      A,64H
        LD      C,A
        LD      B,8
DIV1:   ADD     HL,HL
        LD      A,H
        SUB     C
        JP      C,DIV2
        LD      H,A
        INC     L
DIV2:   DEC     B
        JP      Z,DIVR
        JP      DIV1
DIVR:   EX      DE,HL
        RET
GETP:   LD      C,32            ;GET A AVERAGE DATUM FOR pH
        CALL    GEPPO
        LD      IX,(0DE22H)
        LD      (IX+0H),E
        INC     IX
        LD      (IX+0H),D
        INC     IX
        LD      A,0C9H
        LD      (0DE22H),IX
        LD      C,35            ;GET A AVERAGE DATUM FOR OXYGEN
        CALL    GEPPO
        LD      IX,(0DE24H)
        LD      (IX+0H),E
        INC     IX
        LD      (IX+0H),D
        INC     IX
        LD      A,0C9H
        LD      (0DE24H),IX
        LD      C,30            ;GET A AVERAGE DATUM FOR TEMPERATURE
        CALL    GEPPO
        LD      IX,(0DE26H)
        LD      (IX+0H),E
        INC     IX
        LD      (IX+0H),D
        INC     IX
        LD      A,0C9H
        LD      (0DE26H),IX
        LD      C,34            ;GET A AVERAGE DATUM FOR FOAM SYSTEM
        CALL    GEPPO
        LD      IX,(0DE28H)
        LD      (IX+0H),E
        INC     IX
        LD      (IX+0H),D
        INC     IX
        LD      A,0C9H
        LD      (0DE28H),IX
        LD      C,31            ;GET A AVERAGE DATUM FOR ALKALI SYSTEM
        CALL    GEPPO
        LD      IX,(0DE2AH)
```

```
        LD      (IX+0H),E
        INC     IX
        LD      (IX+0H),D
        INC     IX
        LD      A,0C9H
        LD      (0DE2AH),IX
        RET

; DISPLAYS THE AXIS X AND Y ON THE VIDEO

TRSRR:  LD      DE,(0DE2CH)
        LD      A,(DE)
        INC     DE
        LD      (0DE2CH),DE
        LD      DE,(0DE2EH)
        LD      (DE),A
        INC     DE
        LD      (0DE2EH),DE
        CP      0C9H
        JP      Z,TR1
        JP      TRSRRA
TR1:    LD      DE,0DE50H
        LD      (0DE2EH)
        LD      DE,00800H
        LD      (0DE2CH),DE
DISPLAY:LD      C,0CH
        CALL    STOUT

; DISPLAYS DATA FROM A SPECIFIC PARAMETER


CONTD:  LD      C,0AH
        CALL    STOUT
        LD      C,0DH
        CALL    STOUT
        LD      C,'P'
        CALL    STOUT
        LD      C,'r'
        CALL    STOUT
        LD      C,'='
        CALL    STOUT
        CALL    STIN
        LD      A,'P'
        CP      C
        JP      Z,PHD           ;pH
        LD      A,'O'
        CP      C
        JP      Z,OXD           ;OXYGEN
        LD      A,T
        CP      C
        JP      Z,TEMPD         ;TEMPERATURE
        LD      A,'B'
        CP      C
        JP      Z,BIOMD         ;BIOMASS
        LD      A,'M'
        CP      C
        JP      Z,MOTD          ;ALKALI SYSTEM
        JP      INICIO
```

```
;-------------------------------------------------;
; PUT LABELS OF A GRAPH OF A SPECIFIC PARAMETER   ;
;-------------------------------------------------;

PHD:    LD      DE,(0DD00H)
        LD      (0DE32H),DE     ;INITIAL ADDRESS OF pH DATA ON RAM
        LD      DE,(0DD22H)
        LD      (0DE30H),DE     ;FINAL ADDRESS OF pH DATA ON RAM
        LD      DE,(0DD02H)
        LD      (0DE53H),DE     ;LOAD MEMORY WITH 'p' AND 'H'
        LD      DE,(0DD04H)
        LD      (0DE5BH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      DE,(0DD06H)
        LD      (0DE5DH),DE     ;LOAD MEMORY WITH '.' AND '0'
        LD      DE,(0DD08H)
        LD      (0DE9AH),DE     ;LOAD MEMORY WITH '0' AND '7'
        LD      DE,(0DD0AH)
        LD      (0DE9CH),DE     ;LOAD MEMORY WITH '.' AND '0'
        LD      DE,(0DD0CH)
        LD      (0DEE2H),DE     ;LOAD MEMORY WITH '1' AND '4'
        LD      DE,(0DD0EH)
        LD      (0DEE4H),DE     ;LOAD MEMORY WITH '.' AND '0'
        LD      DE,(0DD0EH)
        CALL    TABLE           ;PREPARE THE GRAPH
        CALL    ACPOT           ;FIND THE SCALE
        RET
OXD:    LD      DE,(0DD10H)
        LD      (0DE32H),DE     ;INITIAL ADDRESS OF OXYGEN DATA ON RAM
        LD      DE,(0DD24H)
        LD      (0DE30H),DE     ;FINAL ADDRESS OF OXYGEN DATA ON RAM
        LD      DE,(0DD12H)
        LD      (0DE52H),DE     ;LOAD MEMORY WITH 'O' AND 'X'
        LD      DE,(0DD14H)
        LD      (0DE54H),DE     ;LOAD MEMORY WITH '.' AND '%'
        LD      DE,(0DD16H)
        LD      (0DE5AH),DE     ;LOAD MEMORY WITH ' ' AND '0'
        LD      DE,(0DD18H)
        LD      (0DE5CH),DE     ;LOAD MEMORY WITH '0' AND '.'
        LD      DE,(0DD1AH)
        LD      (0DE5EH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      DE,(0DD1CH)
        LD      (0DE9AH),DE     ;LOAD MEMORY WITH '5' AND '0'
        LD      DE,(0DD1EH)
        LD      (0DE9CH),DE     ;LOAD MEMORY WITH '.' AND '0'
        LD      DE,(0DD20H)
        LD      (0DEE2H),DE     ;LOAD MEMORY WITH '1' AND '0'
        LD      DE,(0DD22H)
        LD      (0DEE4H),DE     ;LOAD MEMORY WITH '0' AND '.'
        CALL    TABLE           ;PREPARE THE GRAPH
        CALL    ACPOT           ;FIND THE SCALE
        RET
TEMP:   LD      DE,(0DD24H)
        LD      (0DE32H),DE     ;INITIAL ADDRESS OF TEMP. DATA ON RAM
        LD      DE,(0DE26H)
        LD      (0DE30H),DE     ;LOAD MEMORY WITH 'T' AND 'e'
        LD      DE,(0DD26H)
        LD      (0DE51H),DE     ;LOAD MEMORY WITH 'm' AND 'p'
        LD      DE,(0DD28H)
        LD      (0DE53H),DE     ;LOAD MEMORY WITH '.' AND 'o'
```

```
        LD      DE,(0DD2AH)
        LD      (0DE55H),DE     ;LOAD MEMORY WITH 'C' AND ' '
        LD      A,(0DD2CH)
        LD      (0DE57H),DE     ;LOAD MEMORY WITH ' ' AND '0'
        LD      DE,(0DD2EH)
        LD      (0DE5AH),DE     ;LOAD MEMORY WITH '0' AND '.'
        LD      DE,(0DD30H)
        LD      (0DE5CH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      DE,(0DD32H)
        LD      (0DE5EH),DE     ;LOAD MEMORY WITH '5' AND '0'
        LD      DE,(0DD34H)
        LD      (0DE9AH),DE     ;LOAD MEMORY WITH '.' AND '0'
        LD      DE,(0DD36H)
        LD      (0DE9CH),DE     ;LOAD MEMORY WITH '1' AND '0'
        LD      DE,(0DD38H)
        LD      (0DEE2H),DE     ;LOAD MEMORY WITH '0' AND '.'
        LD      DE,(0DD3AH)
        LD      (0DEE4H),DE
        CALL    TABLE           ;PREPARE THE GRAPH
        CALL    ACPOT           ;FIND THE SCALE
        RET
BIOMD:  LD      DE,(0DD3CH)
        LD      (0DE32H),DE     ;INITIAL ADDRESS OF TEMP. DATA
        LD      DE,(0DE2AH)
        LD      (0DE30H),DE     ;FINAL ADDRESS OF TEMP.DATA
        LD      DE,(0DD3EH)
        LD      (0DE51H),DE     ;LOAD MEMORY WITH 'B' AND '.'
        LD      DE,(0DD40H)
        LD      (0DE53H),DE     ;LOAD MEMORY WITH '6' AND '5'
        LD      DE,(0DD42H)
        LD      (0DE55H),DE     ;LOAD MEMORY WITH '0' AND 'n'
        LD      A,(0DD44H)
        LD      (0DE57H),A      ;LOAD MEMORY WITH 'm'
        LD      DE,(0DD45H)
        LD      (0DE5AH),DE     ;LOAD MEMORY WITH ' ' AND '0'
        LD      DE,(0DD47H)
        LD      (0DE5CH),DE     ;LOAD MEMORY WITH '0' AND '.'
        LD      DE,(0DD49H)
        LD      (0DE5EH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      DE,(0DD4BH)
        LD      (0DE9AH),DE     ;LOAD MEMORY WITH '5' AND '0'
        LD      DE,(0DD4DH)
        LD      (0DE9CH),DE     ;LOAD MEMORY WITH '.' AND '0'
        LD      DE,(0DD4FH)
        LD      (0DEE2H),DE     ;LOAD MEMORY WITH '1' AND '0'
        LD      DE,(0DD51H)
        LD      (0DEE4H),DE     ;LOAD MEMORY WITH '0' AND '.'
        CALL    TABLE           ;PREPARE THE GRAPH
        CALL    ACPOT           ;FIND THE SCALE
        RET
MOTD:   LD      DE,(0DD53H)
        LD      (0DE32H),DE     ;INITIAL ADDRESS OF TEMP. DATA
        LD      DE,(0DE28H)
        LD      (0DE30H),DE     ;FINAL ADDRESS OF TEMP.DATA
        LD      DE,(0DD55H)
        LD      (0DE51H),DE     ;LOAD MEMORY WITH 'A' AND 'K'
        LD      DE,(0DD57H)
        LD      (0DE53H),DE     ;LOAD MEMORY WITH 'x' AND '1'
        LD      DE,(0DD59H)
        LD      (0DE55H),DE     ;LOAD MEMORY WITH '0' AND '0'
```

```
        LD      DE,(0DD5BH)
        LD      (0DE5AH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      DE,(0DD5DH)
        LD      (0DE5CH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      DE,(0DD5FH)
        LD      (0DE9AH),DE     ;LOAD MEMORY WITH '0' AND '5'
        LD      DE,(0DD61H)
        LD      (0DE9CH),DE     ;LOAD MEMORY WITH '0' AND '0'
        LD      A,(0DD62H)
        LD      (0DEE4H),A      ;LOAD MEMORY WITH '1'
        CALL    TABLE           ;PREPARE THE GRAPH
        CALL    ACPOT           ;FIND THE SCALE
        RET

;ORGANISES THE GRAPH OF THE RESPECTIVE PARAMETER

TABLE:  LD      HL,0DE50H
TABLO:  LD      A,(HL)
        LD      C,A
        CALL    STOUT
        CP      0C9H
        JP      Z,FIM
        INC     HL
        JP      TABLO
FIM:    RET

;FIND THE SCALE

ACPOT:  LD      A,(0DE31H)
        LD      HL,0DE33H
        CP      (HL)
        JP      Z,FIM
        INC     HL
        JP      TABLO
FIM:    RET
ACPOT:  LD      A,(0DE31H)
        LD      HL,0DE33H
        CP      (HL)
        JP      Z,ACPL1
        INC     (HL)
ACPL1:  CALL    CORCT
        LD      (0DE36H),A
        LD      A,(0DE35H)
        LD      C,A
        LD      DE,(0DE36H)
        EX      DE,HL
        LD      B,8
ALOOP2: ADD     HL,HL
        LD      A,H
        SUB     C
        JP      C,ALOOP1
        LD      H,A
        INC     L
ALOOP1: DEC     B
        JP      Z,ADIVR
        JP      ALOOP2
ADIVR:  LD      A,(0DE42H)
        ADD     A,L
        LD      (0DE38H),A
        CALL    POINT
```

```
        LD      HL,0DE30H
        LD      A,(HL)
        LD      HL,0DE32H
        CP      (HL)
        JP      Z,FINI
        LD      HL,0DE32H
        INC     (HL)
        INC     (HL)
        LD      HL,0DE34H
        INC     (HL)
        LD      A,(0DE34H)
        CP      047H
        JP      Z,FINI
        JP      ACPL1
FINI:   LD      A,00H
        LD      (0DE34H),A
        RET
POINT:  LD      B,06H
LOOPX:  LD      C,20
        CALL    STOUT
        DEC     B
        JP      Z,LOOPY
        JP      LOOPX
LOOPY:  LD      IX,0DE38H
        LD      B,(IX+0H)
LOOP2P: LD      C,01AH
        CALL    STOUT
        DEC     B
        JP      Z,DISP
        JP      LOOP2P
DISP:   LD      C,'*'
        CALL    STOUT
        LD      B,06
LOOP3P: LD      C,08
        CALL    STOUT
        DEC     B
        JP      Z,END
        JP      LOOP3P
END:    LD      C,00AH
        CALL    STOUT
        DEC     B
        JP      Z,ENDT
        JP      LOOP4P
ENDT:   RET
INTF:   OUT     (C),A
        NOP
        NOP
        NOP
        NOP
        NOP
        LD      D,0FFH
STA:    DEC     D
        JP      NZ,STA
        IN      A,(C)
        RET
CONT1:  LD      C,'#'
        CALL    STOUT
        CALL    CONTROL
        RET
CROFF:  LD      A,0FFH
```

```
        OUT     (080H),A
        CALL    INICIO
        RET
GETP1:  LD      C,'0'
        CALL    STOUT
        LD      C,'K'
        CALL    STOUT
        LD      C,'!'
        CALL STOUT
        CALL    GETP
        RET
```

```
0000 31 FF DD 21 FF DC 3E 7A D3 11 3E 35 D3 11 3E 7A
0010 D3 51 3E 35 D3 51 3E 01 32 FF DC 0E 0A CD 74 00
0020 0E 0D CD 74 00 0E 3E CD 74 00 CD A3 00 79 FE 4B
0030 CA B4 02 79 FE 56 CA B2 04 79 FE 46 CA D3 02 79
0040 FE 41 CA 80 01 79 FE 43 CA 3A 01 79 FE 44 CA FC
0050 00 79 FE 45 CA 6B 01 79 FE 4C CA 44 02 79 FE 49
0060 CA 1B 00 79 FE 0D CA 1B 00 CD 74 00 0E 3F CD 74
0070 00 C3 1B 00 3A FF DC EE 01 CA 98 00 DB 51 E6 02
0080 CA 7C 00 DB 50 FE 02 C2 7C 00 DB 51 E6 01 CA 8A
0090 00 79 D3 50 CD 98 00 C9 DB 11 E6 01 CA 74 00 79
00A0 D3 10 C9 DB 11 E6 02 CA A3 00 DB 10 4F CD 74 00
00B0 C9 DB 11 E6 02 C8 DB 10 4F CD 74 00 C9 CD A3 00
00C0 79 FE 30 FA 69 00 79 FE 47 F2 69 00 CD EA 00 C5
00D0 07 07 07 07 47 CD A3 00 79 FE 30 FA 69 00 79 FE
00E0 47 F2 69 00 CD EA 00 80 C1 C9 79 FE 40 FA F8 00
00F0 79 E6 0F C6 09 C3 FB 00 79 E6 0F C9 CD F1 01 0E
0100 2C CD 74 00 CD BD 00 57 CD BD 00 5F CD A3 00 FE
0110 0D C2 69 00 06 10 CD FF 01 CD E8 01 0E 20 CD 74
0120 00 4E CD BB 01 7A AC C2 32 01 7B AD C2 32 01 C3
0130 1B 00 23 05 CA 0A 02 C3 1C 01 CD F1 01 CD A3 00
0140 FE 0D C2 69 00 06 10 CD FF 01 CD E8 01 0E 20 CD
0150 74 00 CD A3 00 FE 0D CA 66 01 CD C0 00 77 23 05
0160 CA 45 01 C3 4D 01 36 C9 C3 1B 00 CD F1 01 CD A3
0170 00 FE 0D C2 1B 00 F5 11 7C 01 D5 E9 F1 C3 1B 00
0180 CD F1 01 CD A3 00 FE 0D C2 69 00 06 10 CD FF 01
0190 CD E8 01 0E 20 CD 74 00 4E CD BB 01 CD A3 00 FE
01A0 20 CA B3 01 79 FE 0D CA 1B 00 CD C0 00 77 0E 20
01B0 CD 74 00 23 05 C2 8B 01 C3 98 01 D5 C5 79 59 16
01C0 02 E6 F0 0F 0F 0F 0F 47 FE 0A 78 FA D4 01 78 C6
01D0 37 C3 D6 01 C6 30 4F CD 74 00 15 CA E5 01 7B E6
01E0 0F 47 C3 C8 01 C1 D1 C9 4C CD BB 01 4D CD BB 01
01F0 C9 0E 20 CD 74 00 CD BD 00 67 CD BD 00 6F C9 0E
0200 0A CD 74 00 0E 0D CD 74 00 C9 CD B1 00 FE 13 CA
0210 1B 02 79 FE 03 CA 1B 00 C3 14 01 CD A3 00 FE 11
0220 CA 14 01 79 FE 03 CA 1B 00 C3 1B 02 F5 DB 11 E6
0230 02 CA 41 02 DB 10 FE 03 C2 41 02 F1 33 33 C3 1B
0240 00 F1 FB C9 0E 0C CD 74 00 0E 56 CD 74 00 0E 3D
0250 CD 74 00 CD A3 00 3E 50 B9 CA 77 02 3E 4F B9 CA
0260 81 02 3E 54 B9 CA 8B 02 3E 42 B9 CA 95 02 3E 4D
0270 B9 CA 9F 02 C3 1B 00 ED 5B 22 DE 21 00 C0 C3 A6
0280 02 ED 5B 24 DE 21 00 C4 C3 A6 02 ED 5B 26 DE 21
0290 00 C8 C3 A6 02 ED 5B 2A DE 21 00 D0 C3 A6 02 ED
02A0 5B 28 DE 21 Q0 CC 3E 00 32 FF DC CD 14 01 3E 01
02B0 CD 74 00 C9 3E 00 D3 4E 3E 00 D3 44 D3 45 D3 46
02C0 D3 47 D3 48 D3 49 D3 4A D3 4B D3 4C D3 4D 3E FF
02D0 D3 4E 00 DB 45 0F 0F 0F 0F E6 3F FE 3F CA D3 02
02E0 32 04 DE CD 5D 03 CD B1 00 FE 03 CA 1B 00 3A 04
02F0 DE E6 01 21 00 DE BE CA FD 02 C3 D3 02 3A 00 DE
0300 EE 01 32 00 DE 21 01 DE 3A 02 DE BE CA 13 03 34
0310 C3 D3 02 CD 28 04 3E 00 32 01 DE C3 D3 02 ED 5B
0320 32 DE 1A 4F 3A 44 DE 91 C9 00 00 00 00 00 00 00
0330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0350 00 00 00 00 00 00 00 00 00 00 00 00 C9 0E 37 CD
0360 D8 03 3A 1C DE BB FA 75 03 3E FF D6 10 D3 80 32
0370 FC DF C3 7C 03 3E FF D3 80 32 FC DF 0E 30 CD D8
```

```
0380 03 3A 1D DE BB FA 92 03 3A FC DF D6 08 D3 80 32
0390 FC DF 0E 32 CD D8 03 3A 1E DE BB FA AB 03 3A FC
03A0 DF D6 80 D3 80 32 FC DF C3 C1 03 0E 32 CD D8 03
03B0 3A 1F DE BB F2 C1 03 3A FC DF D6 01 D3 80 32 FC
03C0 DF 0E 34 CD D8 03 3A 20 DE BB F2 D7 03 3A FC DF
03D0 D6 40 D3 80 32 FC DF C9 21 00 DC 06 64 3E 00 CD
03E0 B3 07 00 77 23 36 00 23 05 CA EF 03 C3 DF 03 ED
03F0 73 FA DF 06 64 31 00 DC 21 00 00 D1 19 05 CA 04
0400 04 C3 FB 03 EB ED 53 F8 DF ED 7B FA DF ED 5B F8
0410 DF EB 3E 64 4F 06 08 29 7C 91 DA 1F 04 67 2C 05
0420 CA 26 04 C3 17 04 EB C9 0E 32 CD D8 03 DD 2A 22
0430 DE DD 73 00 DD 23 DD 72 00 DD 23 3E C9 DD 22 22
0440 DE 0E 37 CD D8 03 DD 2A 24 DE DD 73 00 DD 23 DD
0450 72 00 DD 23 3E C9 DD 77 00 DD 22 24 DE 0E 30 CD
0460 D8 03 DD 2A 26 DE DD 73 00 DD 23 DD 72 00 DD 23
0470 3E C9 DD 77 00 DD 22 26 DE 0E 33 CD D8 03 DD 2A
0480 28 DE DD 73 00 DD 23 DD 72 00 DD 23 3E C9 DD 77
0490 00 DD 22 28 DE 0E 31 CD D8 03 DD 2A 2A DE DD 73
04A0 00 DD 23 DD 72 00 DD 23 3E C9 DD 77 00 DD 22 2A
04B0 DE C9 ED 5B 2C DE 1A 13 ED 53 2C DE ED 5B 2E DE
04C0 12 13 ED 53 2E DE FE C9 CA CE 04 C3 B2 04 11 50
04D0 DE ED 53 2E DE 11 00 08 00 ED 53 2C DE 0E 0C CD
04E0 74 00 0E 0A CD 74 00 0E 0D CD 74 00 0E 50 CD 74
04F0 00 0E 72 CD 74 00 0E 3D CD 74 00 CD A3 00 3E 50
0500 B9 CA 1F 05 3E 4F B9 CA 6E 05 3E 54 B9 CA CD 05
0510 3E 42 B9 CA 3A 06 3E 4D B9 CA A7 06 C3 1B 00 ED
0520 5B 00 DD ED 53 32 DE ED 5B 22 DE ED 53 30 DE ED
0530 5B 02 DD ED 53 53 DE ED 5B 04 DD ED 53 5B DE ED
0540 5B 06 DD ED 53 5D DE ED 5B 08 DD ED 53 9A DE ED
0550 5B 0A DD ED 53 9C DE ED 5B 0C DD ED 53 E2 DE ED
0560 5B 0E DD ED 53 E4 DE CD FC 06 CD 0E 07 C9 ED 5B
0570 10 DD ED 53 32 DE ED 5B 24 DE ED 53 30 DE ED 5B
0580 12 DD ED 53 52 DE ED 5B 14 DD ED 53 54 DE ED 5B
0590 16 DD ED 53 5A DE ED 5B 18 DD ED 53 5C DE ED 5B
05A0 1A DD ED 53 5E DE ED 5B 1C DD ED 53 9A DE ED 5B
05B0 1E DD ED 53 9C DE ED 5B 20 DD ED 53 E2 DE ED 5B
05C0 22 DD ED 53 E4 DE CD FC 06 CD 0E 07 C9 ED 5B 24
05D0 DD ED 53 32 DE ED 5B 26 DE ED 53 30 DE ED 5B 26
05E0 DD ED 53 51 DE ED 5B 28 DD ED 53 53 DE ED 5B 2A
05F0 DD ED 53 55 DE 3A 2C DD 32 57 DE ED 5B 2E DD ED
0600 53 5A DE ED 5B 30 DD ED 53 5C DE ED 5B 32 DD ED
0610 53 5E DE ED 5B 34 DD ED 53 9A DE ED 5B 36 DD ED
0620 53 9C DE ED 5B 38 DD ED 53 E2 DE ED 5B 3A DD ED
0630 53 E4 DE CD FC 06 CD 0E 07 C9 ED 5B 3C DD ED 53
0640 32 DE ED 5B 2A DE ED 53 30 DE ED 5B 3E DD ED 53
0650 51 DE ED 5B 40 DD ED 53 53 DE ED 5B 42 DD ED 53
0660 55 DE 3A 44 DD 32 57 DE ED 5B 45 DD ED 53 5A DE
0670 ED 5B 47 DD ED 53 5C DE ED 5B 49 DD ED 53 5E DE
0680 ED 5B 4B DD ED 53 9A DE ED 5B 4D DD ED 53 9C DE
0690 ED 5B 4F DD ED 53 E2 DE ED 5B 51 DD ED 53 E4 DE
06A0 CD FC 06 CD 0E 07 C9 ED 5B 53 DD ED 53 32 DE ED
06B0 5B 28 DE ED 53 30 DE ED 5B 55 DD ED 53 51 DE ED
06C0 5B 57 DD ED 53 53 DE ED 5B 59 DD ED 53 55 DE ED
06D0 5B 5B DD ED 53 5A DE ED 5B 5D DD ED 53 5C DE ED
06E0 5B 5F DD ED 53 9A DE ED 5B 61 DD E D 53 9C DE 3A
06F0 62 DD 32 E4 DE CD FC 06 CD 0E 07 C9 21 50 DE 7E
```

```
0700 4F CD 74 00 FE C9 CA 0D 07 23 C3 FF 06 C9 3A 31
0710 DE 21 33 DE BE CA 19 07 34 CD 1E 03 00 00 32 36
0720 DE 3A 35 DE 4F ED 5B 36 DE EB 06 08 29 7C 91 00
0730 DA 35 07 67 2C 05 CA 3C 07 C3 2C 07 3A 42 DE 85
0740 32 38 DE CD 6B 07 21 30 DE 7E 21 32 DE BE CA 65
0750 07 21 32 DE 34 34 21 34 DE 34 3A 34 DE FE 47 CA
0760 65 07 C3 19 07 3E 00 32 34 DE C9 06 06 0E 20 CD
0770 74 00 05 CA 79 07 C3 6D 07 DD 21 38 DE DD 46 00
0780 0E 1A CD 74 00 05 CA 8C 07 C3 80 07 0E 2A CD 74
0790 00 06 06 0E 08 CD 74 00 05 CA 9F 07 C3 93 07 DD
07A0 21 38 DE DD 46 00 0E 0A CD 74 00 05 CA B2 07 C3
07B0 A6 07 C9 ED 79 00 00 00 00 00 16 FF 15 C2 BC 07
07C0 ED 78 C9 00 00 00 00 00 00 00 00 00 00 00 00 00
07D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
07E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
07F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0800 0C 20 20 20 20 20 20 20 0A 0D 20 20 20 20 20 20
0810 49 0A 0D 20 20 20 20 20 20 49 0A 0D 20 20 20 20
0820 20 20 49 0A 0D 20 20 20 20 20 20 49 0A 0D 20 20
0830 20 20 20 20 49 0A 0D 20 20 20 20 20 20 49 0A 0D
0840 20 20 20 20 20 20 49 0A 0D 20 20 20 20 20 20 49
0850 0A 0D 20 20 20 20 20 20 49 0A 0D 20 20 20 20 20
0860 20 49 0A 0D 20 20 20 20 20 20 49 0A 0D 20 20 20
0870 20 20 20 49 0A 0D 20 20 20 20 20 20 49 0A 0D 20
0880 20 20 20 20 20 49 0A 0D 20 20 20 20 20 20 49 0A
0890 0D 20 20 20 20 20 20 49 0A 0D 20 20 20 20 20 20
08A0 49 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F
08B0 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F
08C0 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F
08D0 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F
08E0 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 0D 20 20 20 20 20
08F0 20 20 30 20 20 31 20 20 32 20 20 33 20 20 34 20
0900 20 35 20 20 36 20 20 37 20 20 38 20 20 39 20 31
0910 30 20 31 31 20 31 32 20 31 33 20 31 34 20 31 35
0920 20 31 36 20 31 37 20 31 38 20 31 39 20 32 30 20
0930 32 31 20 32 32 20 32 33 20 32 34 0D 20 20 20 20
0940 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0950 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0960 20 20 20 20 54 20 69 20 6D 20 65 20 28 68 6F 75
0970 72 73 29 0D 0A 0A C9 00 00 00 00 00
```

1 - Set point data of parameters used in the fermenter.

The set point datum (XX) from each parameter (hexadecimal) should be found using the table 1a in the end of this appendix (conversion decimal to hexadecimal, e.g the temperature of a experiment is 30.2°C, then the address DE1D is 4D) the datum can be written in the address described in the table 1 below by using the subroutine C or A.

Table 1. Set-point addresses of values for the fermenter parameters.

Address	Content	Parameters
DE1C	XX	OXYGEN
DE1D	XX	TEMPERATURE
DE1E	XX	ACID pH
DE1F	XX	ALKALI pH
DE20	XX	FOAM (constant 10H)

Table 1a - Conversion of parameters set point from decimal to hexadecimal

Decimal	Hexadecimal	Parameters	
		Temperature Biomass Oxygen Foam	pH
1	01	.3921569	.0549017
2	02	.7843138	.1098039
3	03	1.176471	.1647059
4	04	1.568628	.2196079
5	05	1.960784	.2745098
6	06	2.352941	.3294118
7	07	2.745098	.3843137
8	08	3.137255	.4392157
9	09	3.529412	.4941177
10	0A	3.921569	.5490196
11	0B	4.313726	.6039216
12	0C	4.705883	.6588236

13	0D	5.098039	.7137255
14	0E	5.490196	.7686275
15	OF	5.882353	.8235295
16	10	6.27451	.8784314
17	11	6.666667	.9333333
18	12	7.058824	.9882353
19	13	7.450981	1.043137
20	14	7.843138	1.098039
21	15	8.235295	1.152941
22	16	8.627451	1.207843
23	17	9.019608	1.262745
24	18	9.411765	1.317647
25	19	9.803922	1.372549
26	1A	10.19608	1.427451
27	1B	10.58824	1.482353
28	1C	10.98039	1.537255
29	1D	11.37255	1.592157
30	1E	11.76471	1.647059
31	1F	12.15686	1.701961
32	20	12.54902	1.756863
33	21	12.94118	1.811765
34	22	13.33333	1.866667
35	23	13.72549	1.921569
36	24	14.11765	1.976471
37	25	14.5098	2.031373
38	26	14.90196	2.086275
39	27	15.29412	2.141177
40	28	15.68628	2.196078
41	29	16.07843	2.250981
42	2A	16.47059	2.305883
43	2B	16.86275	2.360784
44	2C	17.2549	2.415686
45	2D	17.64706	2.470588
46	2E	18.03922	2.52549
47	2F	18.43137	2.580392
48	30	18.82353	2.635294
49	31	19.21569	2.690196
50	32	19.60784	2.745098
51	33	20	2.8
52	34	20.39216	2.854902
53	35	20.78431	2.909804
54	36	21.17647	2.964706
55	37	21.56863	3.019608
56	38	21.96079	3.07451
57	39	22.35294	3.129412
58	3A	22.7451	3.184314
59	3B	23.13726	3.239216
60	3C	23.52941	3.294118
61	3D	23.92157	3.34902
62	3E	24.31373	3.403922
63	3F	24.70588	3.458824
64	40	25.09804	3.513726
65	41	25.4902	3.568627
66	42	25.88235	3.62353
67	43	26.27451	3.678431
68	44	26.66667	3.733333
69	45	27.05883	3.788235
70	46	27.45098	3.843137
71	47	27.84314	3.898039
72	48	28.2353	3.952941

73	49	28.62745	4.007843
74	4A	29.01961	4.062745
75	4B	29.41177	4.117647
76	4C	29.80392	4.172549
77	4D	30.19608	4.227451
78	4E	30.58824	4.282353
79	4F	30.98039	4.337255
80	50	31.37255	4.392157
81	51	31.76471	4.447059
82	52	32.15687	4.501961
83	53	32.54902	4.556863
84	54	32.94118	4.611765
85	55	33.33333	4.666667
86	56	33.72549	4.721569
87	57	34.11765	4.776471
88	58	34.50981	4.831373
89	59	34.90196	4.886275
90	5A	35.29412	4.941177
91	5B	35.68628	4.996079
92	5C	36.07843	5.050981
93	5D	36.47059	5.105882
94	5E	36.86275	5.160784
95	5F	37.2549	5.215687
96	60	37.64706	5.270589
97	61	38.03922	5.32549
98	62	38.43137	5.380392
99	63	38.82353	5.435294
100	64	39.21569	5.490196
101	65	39.60785	5.545098
102	66	40	5.6
103	67	40.39216	5.654902
104	68	40.78431	5.709804
105	69	41.17647	5.764706
106	6A	41.56863	5.819608
107	6B	41.96079	5.87451
108	6C	42.35295	5.929412
109	6D	42.7451	5.984314
110	6E	43.13726	6.039216
111	6F	43.52941	6.094118
112	70	43.92157	6.14902
113	71	44.31373	6.203922
114	72	44.70589	6.258824
115	73	45.09804	6.313726
116	74	45.4902	6.368628
117	75	45.88236	6.42353
118	76	46.27451	6.478431
119	77	46.66667	6.533334
120	78	47.05883	6.588236
121	79	47.45098	6.643137
122	7A	47.84314	6.698039
123	7B	48.2353	6.752941
124	7C	48.62746	6.807843
125	7D	49.01961	6.862746
126	7E	49.41177	6.917647
127	7F	49.80392	6.972549
128	80	50.19608	7.027451
129	81	50.58824	7.082353
130	82	50.98039	7.137255
131	83	51.37255	7.192157
132	84	51.76471	7.247059

133	85	52.15687	7.301961
134	86	52.54902	7.356863
135	87	52.94118	7.411765
136	88	53.33334	7.466667
137	89	53.72549	7.521569
138	8A	54.11765	7.576471
139	8B	54.50981	7.631373
140	8C	54.90196	7.686275
141	8D	55.29412	7.741177
142	8E	55.68628	7.796078
143	8F	56.07843	7.850981
144	90	56.47059	7.905883
145	91	56.86275	7.960785
146	92	57.2549	8.015686
147	93	57.64706	8.070588
148	94	58.03922	8.12549
149	95	58.43138	8.180393
150	96	58.82353	8.235295
151	97	59.21569	8.290196
152	98	59.60785	8.345099
153	99	60	8.399999
154	9A	60.39216	8.454902
155	9B	60.78432	8.509804
156	9C	61.17647	8.564706
157	9D	61.56863	8.619608
158	9E	61.96079	8.67451
159	9F	62.35295	8.729412
160	A0	62.7451	8.784313
161	A1	63.13726	8.839215
162	A2	63.52941	8.894118
163	A3	63.92157	8.949019
164	A4	64.31373	9.003921
165	A5	64.70589	9.058824
166	A6	65.09804	9.113726
167	A7	65.4902	9.168628
168	A8	65.88236	9.22353
169	A9	66.27451	9.278431
170	AA	66.66666	9.333333
171	AB	67.05883	9.388235
172	AC	67.45098	9.443137
173	AD	67.84314	9.498039
174	AE	68.2353	9.552941
175	AF	68.62745	9.607843
176	B0	69.01961	9.662744
177	B1	69.41177	9.717647
178	B2	69.80393	9.772549
179	B3	70.19608	9.827451
180	B4	70.58823	9.882353
181	B5	70.9804	9.937255
182	B6	71.37255	9.992157
183	B7	71.76471	10.04706
184	B8	72.15686	10.10196
185	B9	72.54902	10.15686
186	BA	72.94118	10.21176
187	BB	73.33334	10.26667
188	BC	73.7255	10.32157
189	BD	74.11765	10.37647
190	BE	74.50981	10.43137
191	BF	74.90196	10.48628
192	C0	75.29412	10.54118

193	C1	75.68628	10.59608
194	C2	76.07843	10.65098
195	C3	76.47059	10.70588
196	C4	76.86275	10.76078
197	C5	77.25491	10.81569
198	C6	77.64706	10.87059
199	C7	78.03921	10.92549
200	C8	78.43138	10.98039
201	C9	78.82353	11.03529
202	CA	79.21569	11.0902
203	CB	79.60785	11.1451
204	CC	80	11.2
205	CD	80.39216	11.2549
206	CE	80.78432	11.3098
207	CF	81.17647	11.36471
208	D0	81.56863	11.41961
209	D1	81.96079	11.47451
210	D2	82.35295	11.52941
211	D3	82.7451	11.58431
212	D4	83.13725	11.63922
213	D5	83.52941	11.69412
214	D6	83.92157	11.74902
215	D7	84.31373	11.80392
216	D8	84.70589	11.85882
217	D9	85.09804	11.91373
218	DA	85.49019	11.96863
219	DB	85.88236	12.02353
220	DC	86.27451	12.07843
221	DD	86.66667	12.13333
222	DE	87.05882	12.18824
223	DF	87.45098	12.24314
224	E0	87.84314	12.29804
225	E1	88.2353	12.35294
226	E2	88.62745	12.40784
227	E3	89.01961	12.46275
228	E4	89.41177	12.51765
229	E5	89.80392	12.57255
230	E6	90.19608	12.62745
231	E7	90.58823	12.68235
232	E8	90.98039	12.73726
233	E9	91.37255	12.79216
234	EA	91.76471	12.84706
235	EB	92.15686	12.90196
236	EC	92.54902	12.95686
237	ED	92.94118	13.01177
238	EE	93.33334	13.06667
239	EF	93.72549	13.12157
240	F0	94.11764	13.17647
241	F1	94.50981	13.23137
242	F2	94.90196	13.28627
243	F3	95.29412	13.34118
244	F4	95.68628	13.39608
245	F5	96.07843	13.45098
246	F6	96.47059	13.50588
247	F7	96.86275	13.56078
248	F8	97.25491	13.61569
249	F9	97.64706	13.67059
250	FA	98.03921	13.72549
251	FB	98.43138	13.78039
252	FC	98.82353	13.83529

253	FD	99.21569	13.8902
254	FE	99.60784	13.9451
255	FF	100	14

Figure APE6 shows the listing of the program PLODIS.BAS, which utilises the medium resolution graphics facilities in the Advanced Basic language in the IBM microcomputer. Figure APE1 shows a example of how to use it. It is simple and comprehensive to use. Figure APE5 shows the growth curve of the yeast Saccharomyces cerevisiae Σ1278b in bactch culture. To run this program, you just press F2, then choose one of the options.

Figure APE1 - How to run the PLODIS

Press F2, then

DIAGRAM MENU

M : menu routine
E : end Generation
N : new diagram
R : reload diagram
S : store diagram
B : bring diagram to screen
L : label diagram
H : histogram or bar-chart
P : pie-chart
C : continuous graph
D : discrete graph

Press D, then

```
Would you like a line between points (Y/N)? Y
Data are in the disk (Y/N)? N

Number of points ? 5
Type X(1) and Y(1)? 0,1
Type X(2) and Y(2)? 2,3
Type X(3) and Y(3)? 4,6
Type X(4) and Y(4)? 7,9
Type X(5) and Y(5)? 8,10

Lower and Upper bounds for X? 0,10
Lower and Upper bounds for Y? 0,10
Do you want X-axis labelled Y/N ? Y
Do you want Y-axis labelled Y/N ? Y
```

After these procedure the figure APE3 will appear on the video which corresponds to the graph, but without any names on it. In order to get names, it will be necessary to go back to the menu, then select the option L, which has a series of extra drawing possibilities, as described in figure APE2.

Figure APE2 - Labelling Menu

LABELLING MENU

M : menu routine
X : exit labelling
B : bring picture to screen
G : store picture to screen
C : change colours
V : vertical labels
H : horizontal labels
N : narrow labels
P : draw a point
L or K : draw a line or a circle
D : draw a block
T or F : tile-paint or fill a block

Then you just choose a place to put your label, using the arrow keys, then press the key end and write the label. Finally, do not forget to press G <CR> in order to fix you label otherwise you will loose it. Figure APE4 shows the graph of figure APE3 with labels on it.

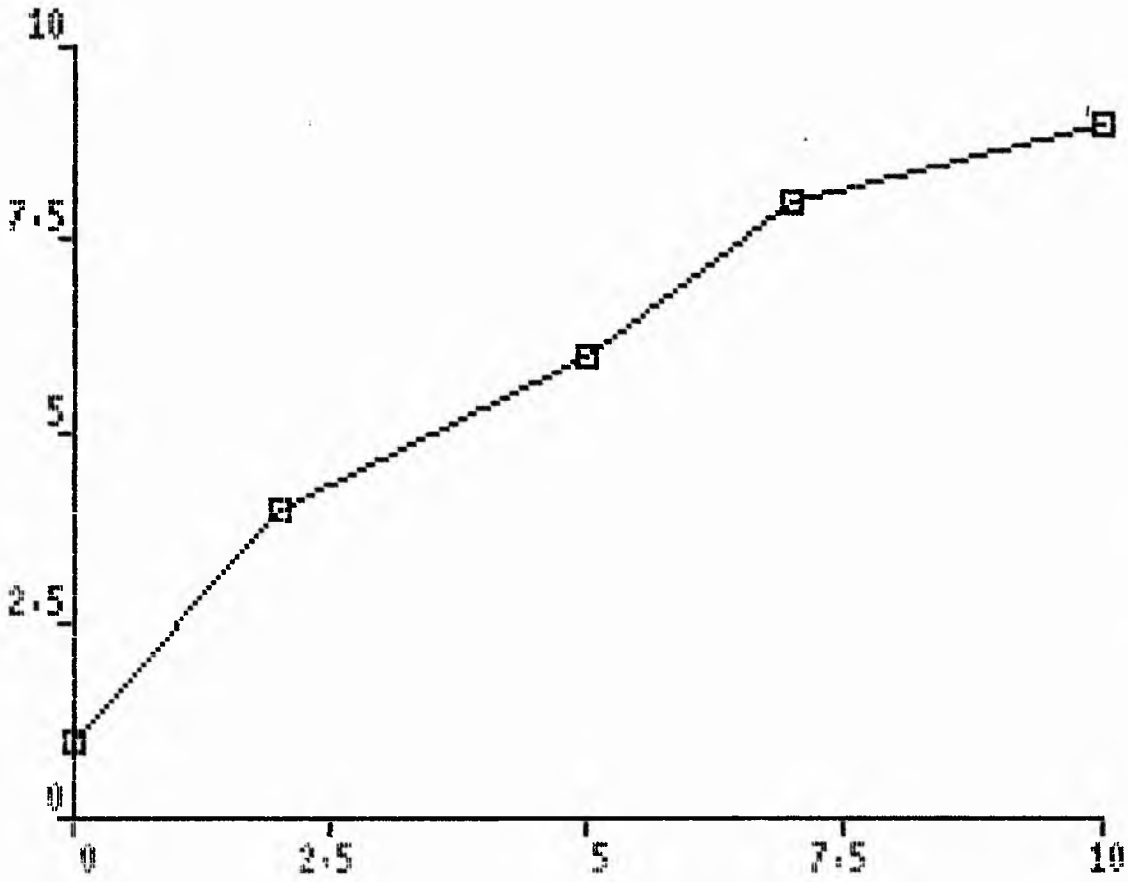


Figure APE3 - Example without labels

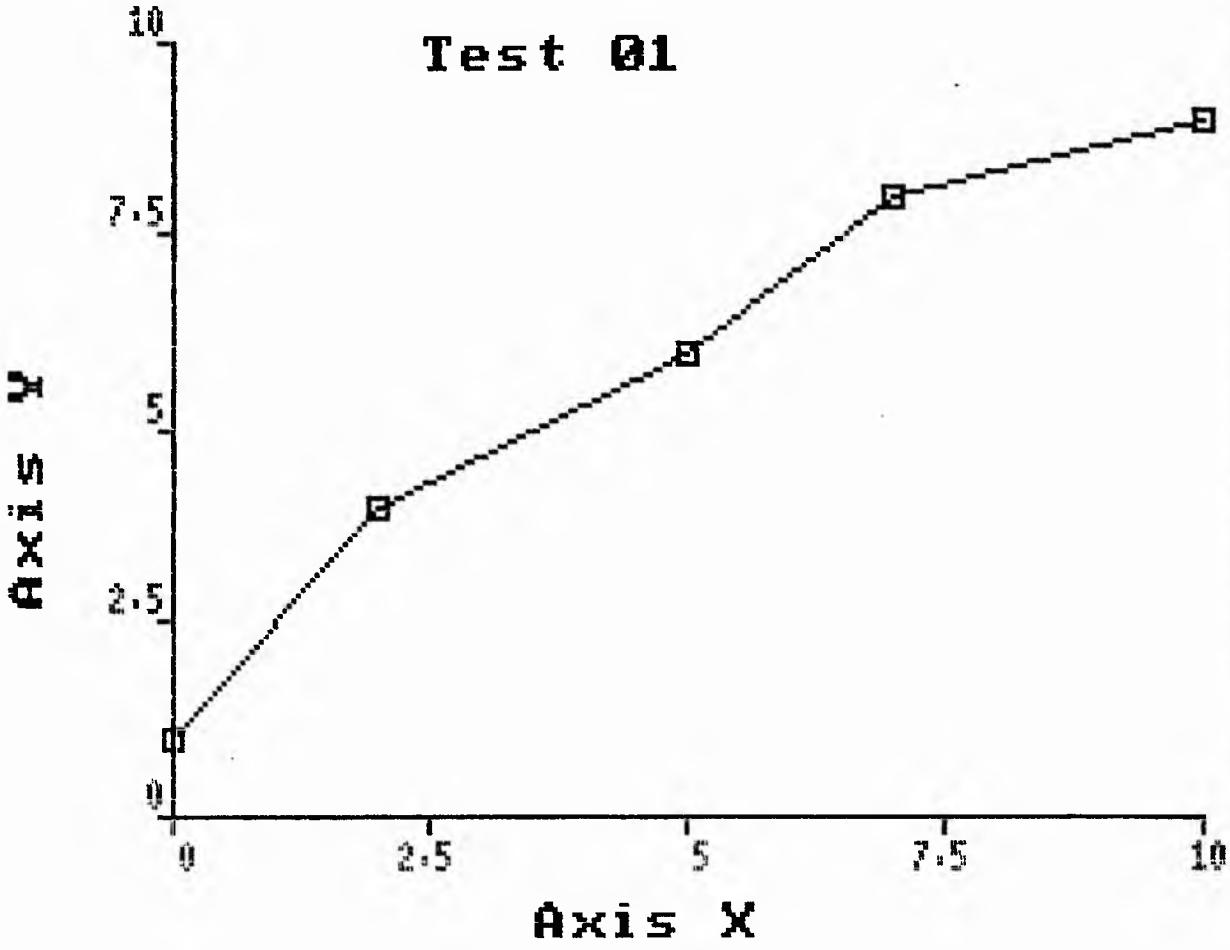


Figure APE4 - figure APE3 with labels

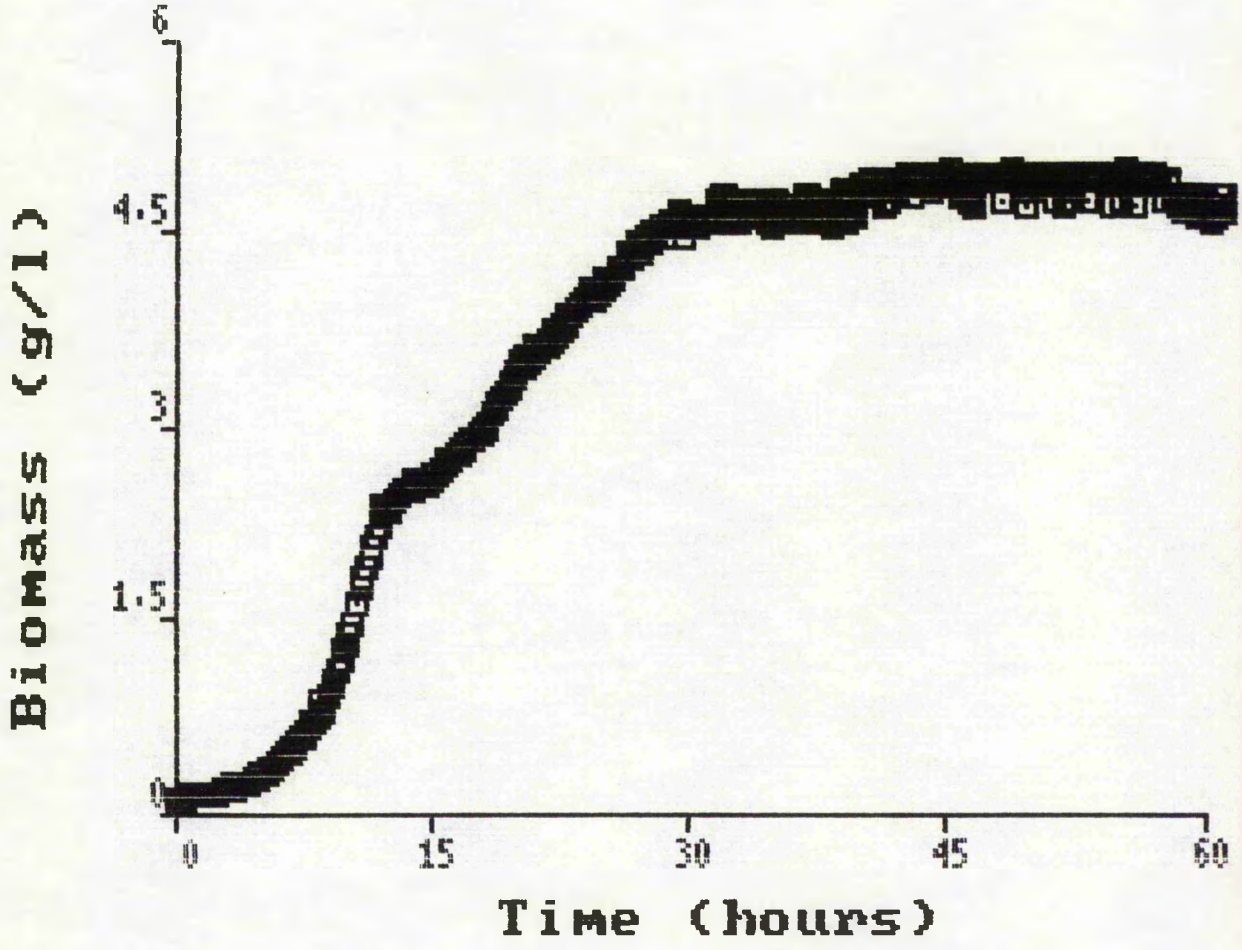


Figure APE5 - Growth curve of the Saccharomyces cerevisae wild type Σ1278b in batch culture.

Figure APE6 - PLODIS program

```
100 ' DATA DIAGRAM PROGRAM
200 CLS : OPTION BASE 1
300 DIM CURSOR%(7),SKREEN%(8002),BLANK%(18)
400 RESTORE 500
500 DATA &HA,&H5,&H4,&H4,&H4055,&H4,&H4
600 FOR I%=1 TO 7 : READ CURSOR%(I%) : NEXT I%
700 CURX%=160 : CURY%=100 : INC%=8
800 GOSUB 12000 'SETUP NARROW CHARACTERS
900 DEF SEG=0
1000 PAGE%=PEEK(127)*&H100+PEEK(126)
1100 ADDRESS%=PEEK(125)*&H100+PEEK(124)
1200 DEF SEG=PAGE% : BLOAD"ROTCHARS.DAT",ADDRESS%
1300 MODE%=1 : ASPECT=1
1400 PALETTE%=6 : BACKGROUND%=1
1500 TRUE%=-1 : FALSE%=0
1600 DIM TILE$(16),BOUND%(16)
1700 BOUND%(1)=2
1800 BOUND%(2)=1
1900 BOUND%(3)=0
2000 BOUND%(4)=3
2100 TILE$(5)=CHR$(&H11)+CHR$(&H44) : BOUND%(5)=2
2200 TILE$(6)=CHR$(&H22)+CHR$(&H88) : BOUND%(6)=1
2300 TILE$(7)=CHR$(&H33)+CHR$(&HCC) : BOUND%(7)=0
2400 TILE$(8)=CHR$(&H66)+CHR$(&H99) : BOUND%(8)=0
2500 TILE$(9)=CHR$(&H77)+CHR$(&HDD) : BOUND%(9)=2
2600 TILE$(10)=CHR$(&HBB)+CHR$(&HEE) : BOUND%(10)=1
2700 TILE$(11)=CHR$(&H0)+CHR$(&HA) : BOUND%(11)=0
2800 TILE$(12)=CHR$(&H0)+CHR$(&H2A)+CHR$(&H2A)+CHR$(&H2A) : BOUND%(12)=
2900 TILE$(13)=CHR$(&H1B)+CHR$(&H6A)+CHR$(&HB1)+CHR$(&HA6) : BOUND%(13)
3000 TILE$(14)=CHR$(&H55)+CHR$(&H11) : BOUND%(14)=0
3100 TILE$(15)=CHR$(&H55)+CHR$(&H66) : BOUND%(15)=1
3200 TILE$(16)=CHR$(&HFF)+CHR$(&HEE) : BOUND%(16)=2
3300 GOSUB 47300 'colour monitor
3400 PAPER%=FOREGROUND% : GOSUB 7900 'new screen
3500 GOSUB 5400 'diagram menu
3600 GOSUB 3900 'diagramm supervisor
3700 GOSUB 48300 ' monochrome monitor
3800 END
3900 ' diagram supervisor routine
4000 GOSUB 6700 ' keyboard input
4100 IF LENKB%=2 THEN  GOTO 4000
4200 IF ASCKB%>90 THEN ASCKB%=ASCKB%-32
4300 IF ASCKB%=ASC("M") THEN GOSUB 5400 ' diagram menu
4400 IF ASCKB%=ASC("N") THEN GOSUB 7500 : GOSUB 5400 ' new diagram
4500 IF ASCKB%=ASC("R") THEN GOSUB 10300 'reload diagram routine
4600 IF ASCKB%=ASC("S") THEN GOSUB 11100 'store diagram routine
4700 IF ASCKB%=ASC("B") THEN PUT (0,0),SKREEN%,PSET
4800 IF ASCKB%=ASC("L") THEN GOSUB 14200 'label routine
4900 IF ASCKB%=ASC("P") THEN GOSUB 35700 ' pie-chart routine
5000 IF ASCKB%=ASC("C") THEN GOSUB 35800 ' continuous graph
```

```
5100 IF ASCKB%=ASC("D") THEN GOSUB 41600 ' discrete graph
5200 IF ASCKB%<>ASC("E") THEN GOTO 4000
5300 RETURN
5400 'diagram menu routine
5500 CLS
5600 LOCATE  2,14 : PRINT "DIAGRAM MENU";
5700 LOCATE  4,10 : PRINT "M : menu routine ";
5800 LOCATE  6,10 : PRINT "E : end generation";
5900 LOCATE  8,10 : PRINT "N : new diagram";
6000 LOCATE 10,10 : PRINT "R : reload diagram";
6100 LOCATE 12,10 : PRINT "S : store diagram ";
6200 LOCATE 14,10 : PRINT "B : bring diagram to screen";
6300 LOCATE 16,10 : PRINT "L : label diagram";
6400 LOCATE 18,10 : PRINT "C : continuous graph";
6500 LOCATE 20,10 : PRINT "D : discrete graph";
6600 RETURN
6700 ' keyboard input routine
6800 '**out** asckb%,lenkb%
6900 'find ascii code asckb$ of key pressed
7000 KB$=INKEY$ : LENKB%=LEN(KB$)
7100 IF LENKB%=2 THEN KB$=RIGHT$(KB$,1)
7200 IF KB$="" THEN  GOTO 7000
7300 ASCKB%=ASC(KB$)
7400 RETURN
7500 'new diagram / clear the screen
7600 CLS : INPUT"ARE YOU SURE?(Y/N) ";F$
7700 IF F$<>"y" AND F$<>"Y" THEN RETURN
7800 INPUT"colour of paper ";PAPER%
7900 LINE (0,0)-(319,199),PAPER%,BF
8000 GET (0,0)-(319,199),SKREEN%
8100 RETURN
8200 ' cursor control
8300 IF MENUONSCREEN% THEN RETURN
8400 PUT (CURX%-2,CURY%-2),CURSOR%,XOR
8500 GOSUB 6700 ' keyboard input
8600 IF ASCKB%=71 THEN  INC%=9-INC% : GOTO 8500
8700 IF ASCKB%=80 THEN MX%=0 : MY%=INC% : GOSUB 9400 : GOTO 8500
8800 IF ASCKB%=72 THEN MX%=0 : MY%=-INC% : GOSUB 9400 : GOTO 8500
8900 IF ASCKB%=75 THEN MX%=-INC% : MY%=0 : GOSUB 9400 : GOTO 8500
9000 IF ASCKB%=77 THEN MX%=INC% : MY%=0 : GOSUB 9400 : GOTO 8500
9100 IF ASCKB%>47 AND ASCKB%<52 THEN PSET (CURX%,CURY%),(ASCKB%-48) XOR
9150 GOTO 8500
9200 IF ASCKB%=79 THEN PUT (CURX%-2,CURY%-2),CURSOR%,XOR : RETURN
9300 GOTO 8500
9400 'move the cursor cross at (curx%,cury%) by (mx%,my%)
9500 NEWX%=CURX%+MX%
9600 IF NEWX%<2 OR NEWX%>317 THEN RETURN
9700 NEWY%=CURY%+MY%
9800 IF NEWY%<2 OR NEWY%>197 THEN RETURN
9900 PUT (CURX%-2,CURY%-2),CURSOR%,XOR
10000 CURX%=NEWX% : CURY%=NEWY%
```

```
10100 PUT (CURX%-2,CURY%-2),CURSOR%,XOR


10200 RETURN
10300 'reload diagram from disk : get  it in screen%
10400 CLS
10500 LOCATE 10,1 : PRINT "Type name of file that holds picture";
10600 LOCATE 11,1 : PRINT"e.g  B:PICCY.PIC";
10700 LOCATE 14,1 : INPUT F$
10800 DEF SEG=&HB800 : BLOAD F$,0
10900 GET (0,0)-(319,199),SKREEN%
11000 RETURN
11100 'store diagram on disk from skreen%
11200 CLS
11300 LOCATE 10,1 : PRINT"Type name of file that will hold picture";
11400 LOCATE 11,1 : PRINT"e.g  B:DIAGRAM.PIC";
11500 LOCATE 14,1 : INPUT F$
11600 CLS : PUT (0,0),SKREEN%
11700 DEF SEG=&HB800 : BSAVE F$,0,&H3FFF
11800 GOSUB 5400 'diagram menu
11900 RETURN
12000 'setup narrow characteres from disk
12100 DIM NARROW%(780),BUFFER%(6)
12200 OPEN "i",#1,"narrow.dat"
12300 I%=0
12400 I%=I%+1
12500 ON ERROR GOTO 12800
12600 INPUT#1,NARROW%(I%)
12700 GOTO 12400
12800 CLOSE
12900 RETURN
13000 'narrow print routine
13100 XP%=X%
13200 FOR I%=1 TO LEN(NUM$)
13300 CHAR$=MID$(NUM$,I%,1)
13400 IF CHAR$>="0" AND CHAR$<="9" THEN CHAR%=(ASC(CHAR$)-48)*6
ELSE IF CHAR$="+" THEN CHAR%=60 ELSE IF CHAR$="-" THEN CHAR%=66
ELSE CHAR%=72
13500 FOR J%=1 TO 6
13600 BUFFER%(J%)=NARROW%(CHAR%+J%)
13700 NEXT J%
13800 PUT (XP%,Y%),BUFFER%
13900 XP%=XP%+4
14000 NEXT I%
14100 RETURN
14200 ' label supervisor routine
14300 FOREGROUND%=3 : BACKGROUND%=3
14400 GOSUB 22800 ' change colours
14500 GOSUB 6700 ' keyboard input
14600 IF LENKB%<>1 THEN GOTO 14500
14700 IF ASCKB%>90 THEN ASCKB%=ASCKB%-32
14800 IF ASCKB%=ASC("M") THEN GOSUB 16400 'label menu
14900 IF ASCKB%=ASC("B") THEN PUT (0,0),SKREEN%,PSET :
MENUONSCREEN%=FALSE%
15000 IF ASCKB%=ASC("G") THEN GET (0,0)-(319,199),SKREEN%
```

```
15100 IF ASCKB%=ASC("C") THEN GOSUB 22200 ' change colours
15200 IF ASCKB%=ASC("V") THEN GOSUB 18100 'vertical label routine
15300 IF ASCKB%=ASC("H") THEN GOSUB 20300 ' horizontal label
15400 IF ASCKB%=ASC("N") THEN GOSUB 27100 'narrow label
15500 IF ASCKB%=ASC("P") THEN GOSUB 8200 'cursor start
15600 IF ASCKB%=ASC("L") THEN GOSUB 25400 'draw a line
15700 IF ASCKB%=ASC("D") THEN GOSUB 23200 'drag a block
15800 IF ASCKB%=ASC("F") THEN GOSUB 24600 'fill a block
15900 IF ASCKB%=ASC("T") THEN GOSUB 28200 'tile-paint
16000 IF ASCKB%=ASC("K") THEN GOSUB 29500 ' drow a circle
16100 IF ASCKB%<>ASC("X") THEN GOTO 14500
16200 GOSUB 5400 'diagram menu
16300 RETURN
16400 'label menu routine
16500 CLS
16600 LOCATE  1,13 : PRINT "LABELLING MENU";
16700 LOCATE  3,10 : PRINT "M : menu routine";
16800 LOCATE  5,10 : PRINT "X : exit labelling";
16900 LOCATE  7,10 : PRINT "B : bring picture to screen";
17000 LOCATE  9,10 : PRINT "G : store picture to screen";
17100 LOCATE 11,10 : PRINT "C : change colours";
17200 LOCATE 13,10 : PRINT "V : vertical labels";
17300 LOCATE 15,10 : PRINT "H : horizontal labels";
17400 LOCATE 17,10 : PRINT "N : narrow labels";
17500 LOCATE 19,10 : PRINT "P : draw a point";
17600 LOCATE 21,5 : PRINT "L or K :draw a line or a circle";
17700 LOCATE 23,10 : PRINT "D : draw a block";
17800 LOCATE 25,5  : PRINT "T or F : tile-paint or fill a block";
17900 MENUONSCREEN%=TRUE%
18000 RETURN
18100 'vertical label routine
18200 IF MENUONSCREEN% THEN RETURN
18300 GOSUB 8200 'cursor control
18400 LABCOL%=INT(CURX%/8)+1
18500 LABROW%=INT(CURY%/8)+1
18600 LINE (0,0)-(319,15),0,BF
18700 LOCATE 1,1 : PRINT "label ?"
18800 LOCATE 2,1 : INPUT LABEL1$
18900 PUT (0,0),SKREEN%,PSET
19000 LENLAB%=LEN(LABEL1$)
19100 FOR LROW%=1 TO LENLAB%
19200 LOCATE LABROW%+1-LROW%,LABCOL%
19300 ASCHAR%=ASC(MID$(LABEL1$,LROW%,1))+128
19400 PRINT CHR$(ASCHAR%);
19500 NEXT LROW%
19600 LINE (0,0)-(319,15),0,BF
19700 CURX%=LABCOL%*8-8 : CURY%=LABROW%*8-8
19800 FOR LROW%=1 TO LENLAB%
19900 PUT (CURX%,CURY%),BLANK%
20000 CURY%=CURY%-8
```

```
20100 NEXT LROW%
20200 RETURN
20300 'horizontal label routine
20400 IF MENUONSCREEN% THEN RETURN
20500 GOSUB 8200 'cursor control
20600 LABCOL%=INT(CURX%/8)+1
20700 LABROW%=INT(CURY%/8)+1
20800 LINE (0,0)-(319,15),0,BF
20900 LOCATE 1,1 : PRINT "label ?"
21000 LOCATE 2,1 : INPUT LABEL$
21100 PUT (0,0),SKREEN%,PSET
21200 LOCATE LABROW%,LABCOL% : PRINT LABEL$;
21300 CURX%=LABCOL%*8-8 : CURY%=LABROW%*8-8
21400 LENLAB%=LEN(LABEL$)
21500 FOR LROW%=1 TO LENLAB%
21600 PUT (CURX%,CURY%),BLANK%
21700 CURX%=CURX%+8
21800 NEXT LROW%
21900 IF CURX%>312 THEN CURX%=312
22000 IF CURY%<8 THEN CURY%=8
22100 RETURN
22200 'change foreground and background text colours
22300 CLS : INPUT "foreground colour ";FOREGROUND%
22400 IF FOREGROUND%<0 OR FOREGROUND%>3 THEN GOTO 22300
22500 INPUT "background colour ";BACKGROUND%
22600 IF BACKGROUND%<0 OR BACKGROUND%>3 THEN GOTO 22300
22700 LINE (90,0)-(7,7),BACKGROUND%,BF
22800 GET (0,0)-(7,7),BLANK%
22900 IF BACKGROUND%<>FOREGROUND% THEN DEF SEG :
 POKE &H4E,FOREGROUND% XOR BACKGROUND%
23000 GOSUB 16400 'label menu
23100 RETURN
23200 'drag a block
23300 IF MENUONSCREEN% THEN RETURN
23400 GOSUB 26100 'define a line
23500 PUT (CURX1%-2,CURY1%-2),CURSOR%
23600 PUT (CURX2%-2,CURY2%-2),CURSOR%
23700 SIZ%=2+INT(2*(ABS(CURX2%-CURX1%)+8)/8)*(ABS(CURY2%-CURY1%)+1)
23800 DIM HOLD%(SIZ%)
23900 GET (CURX1%,CURY1%)-(CURX2%,CURY2%),HOLD%
24000 LINE (CURX1%,CURY1%)-(CURX2%,CURY2%),3-PAPER%,B
24100 GOSUB 8200 'cursor start
24200 LINE (CURX1%,CURY1%)-(CURX2%,CURY2%),PAPER%,BF
24300 PUT (CURX%,CURY%),HOLD%,PSET
24400 ERASE HOLD%
24500 RETURN
24600 'fill a block
24700 IF MENUONSCREEN% THEN RETURN
24800 GOSUB 26100 'define a line
24900 PUT (CURX1%-2,CURY1%-2),CURSOR%
25000 PUT (CURX2%-2,CURY2%-2),CURSOR%
```

```
25100 LINE (CURX1%,CURY1%)-(CURX2%,CURY2%),BACKGROUND%,BF
25200 LINE (CURX1%,CURY1%)-(CURY2%,CURY2%),FOREGROUND%,B
25300 RETURN
25400 'draw aline
25500 IF MENUONSCREEN% THEN RETURN
25600 GOSUB 26100 'define a line
25700 PUT (CURX1%-2,CURY1%-2),CURSOR%
25800 PUT (CURX2%-2,CURY2%-2),CURSOR%
25900 LINE (CURX1%,CURY1%)-(CURX2%,CURY2%),15
26000 RETURN
26100 'define a line
26200 GOSUB 8200 'start cursor
26300 CURX1%=CURX% : CURY1%=CURY%
26400 PUT (CURX%-2,CURY%-2),CURSOR%
26500 IF CURX%>309 THEN CURX%=CURX%-8 ELSE CURX%=CURX%+8
26600 IF CURY%>189 THEN CURY%=CURY%-8 ELSE CURY%=CURY%+8
26700 GOSUB 8200 'start cursor
26800 CURX2%=CURX% : CURY2%=CURY%
26900 PUT (CURX%-2,CURY%-2),CURSOR%
27000 RETURN
27100 'narrow horizontal label
27200 IF MENUONSCREEN% THEN RETURN
27300 GOSUB 8200 'cursor start
27400 LINE (0,0)-(319,15),0,BF
27500 LOCATE 1,1 : PRINT "label ?"
27600 LOCATE 2,1 : INPUT NUM$
27700 PUT (0,0),SKREEN%,PSET
27800 LINE (0,0)-(319,15),0,BF
27900 X%=CURX% : Y%=CURY%
28000 GOSUB 13000 ' narrow labels
28100 RETURN
28200 'tile-paint an area
28300 IF MENUONSCREEN% THEN RETURN
28400 LINE (0,0)-(319,15),0,BF
28500 LOCATE 1,1 : PRINT "tile number ?"
28600 LOCATE 2,1 : INPUT TILE%
28700 PUT (0,0),SKREEN%,PSET
28800 GOSUB 26100 ' define a line
28900 PUT (CURX1%-2,CURY1%-2),CURSOR%
29000 PUT (CURX2%-2,CURY2%-2),CURSOR%
29100 LINE (CURX1%,CURY1%)-(CURX2%,CURY2%),3-BOUND%(TILE%),BF
29200 LINE (CURX1%,CURY1%)-(CURX2%,CURY2%),BOUND%(TILE%),B
29300 IF TILE%>4 THEN PAINT ((CURX1%+CURX2%)/2,(CURY1%+CURY2%)/2),
 TILE$(TILE$),BOUND%(TILE%)
29400 RETURN
29500 'draw a cirlle
29600 IF MENUONSCREEN% THEN RETURN
29700 GOSUB 26100
29800 PUT (CURX1%-2,CURY1%-2),CURSOR%
29900 PUT (CURX2%-2,CURY2%-2),CURSOR%
30000 CIRCLE (CURX1%,CURY1%),ABS((CURX1%-CURY1%)-(CURX2%-CURY2%)),15
```

```
30100 RETURN
30200 ' Bar-chart routine (type 2
30300 '** out ** SKREEN%
30400 CLS
30500 'input data for 3 - d bar chart.
30600 INPUT"Lower bound of vertical range ";YLO
30700 INPUT "Upper bound of vertical range ";YHI
30800 IF YLO >= YHI THEN GOTO 30300
30900 YSCALE=150/(YHI-YLO)
31000 INPUT "Number of bars in each row ",NBAR%
31100 DIM BARVAL(2,NBAR%)
31200 FOR I%=1 TO NBAR%
31300 PRINT "Front and back values for bar ";STR$(I%)
31400 INPUT BARVAL(1,I%),BARVAL(2,I%)
31500 NEXT I%
31600 'draw axes
31700 LINE (0,0)-(319,199),3,BF
31800 LINE (40,25)-(40,175),0
31900 LINE (40,175)-(300,175),0
32000 'place 5 labels on vertical axis.
32100 YV=YLO : YDIF=(YHI-YLO)/4
32200 FOR K%=1 TO 5
32300 'place small diagonal mark on axis
32400 Y%=175-INT((YV-YLO)*YSCALE+.5)
32500 LINE (43,Y%-3)-(37,Y%+3),0
32600 'place narrow label at mark.
32700 NUM$=STR$(INT(YV))
32800 IF YV>=0 THEN NUM$=RIGHT$(NUM$,LEN(NUM$)-1)
32900 IF LEN(NUM$)>5 THEN NUM$="..."
33000 X%=36-4*LEN(NUM$) : Y%=Y%-4
33100 GOSUB 13000 ' narrow characters
33200 YV=YV+YDIF
33300 NEXT K%
33400 GAP=INT(260/NBAR%/4-1) : EXTRA=GAP+(260-4*NBAR%*(GAP+1))/2
33500 FOR ROW%=1 TO 2
33600 XS=40+EXTRA-GAP*(2-ROW%) : YS=175+GAP*(2-ROW%)
33700 FOR I%=1 TO NBAR%
33800 X=XS+4*(GAP+1)*(I%-1)
33900 YPIXEL=YS-YSCALE*(BARVAL(ROW%,I%)-YLO)
34000 LINE (X,YS)-(X+2*GAP,YS),ROW%
34100 LINE -(X+3*GAP,YS-GAP),ROW%
34200 LINE -(X+3*GAP,YPIXEL-GAP),ROW%
34300 LINE -(X+GAP,YPIXEL-GAP),ROW%
34400 LINE -(X,YPIXEL),ROW%
34500 LINE -(X,YS),ROW%
34600 'paint
34700 IF YPIXEL<200 THEN PAINT(X+2*GAP,YPIXEL),3-ROW%,ROW%
 ELSE PAINT (X+2*GAP,199),3-ROW%,ROW%
34800 LINE (X+2*GAP,YS)-(X+2*GAP,YPIXEL),ROW%
34900 LINE (X,YPIXEL)-(X+2*GAP,YPIXEL),ROW%
35000 LINE -(X+3*GAP,YPIXEL-GAP),ROW%
```

```
35100 NEXT I%
35200 IF ROW%=1 THEN GET (0,0)-(319,199),SKREEN% :
 LINE (0,0)-(319,199),3,BF ELSE PUT (0,0),SKREEN%,AND
35300 NEXT ROW%
35400 GET (0,0)-(319,199),SKREEN%
35500 ERASE BARVAL
35600 RETURN
35700 RETURN
35800 'continuous graph .
35900 GOSUB 37500 ' prepare screen and axes
36000 X=(P0Y%-50)/XSCALE+XLO : GOSUB 37200 'find y-value from x
36100 PY%=INT(175-(Y-YLO)*YSCALE)
36200 PSET (P0Y%,PY%),0
36300 FOR PX%=51 TO 270
36400 X=(PX%-49)/XSCALE+XLO :GOSUB 37200 'find y-value
36500 PY%=INT(175-(Y-YLO)*YSCALE)
36600 IF PX%<P0Y% THEN GOTO 36900
36700 IF PX%>P1Y% THEN GOTO 37000
36800 LINE - (PX%,PY%),15
36900 NEXT PX%
37000 GET (0,0)-(319,199),SKREEN%
37100 RETURN
37200 'find y-value from x
37300 Y=-15057.24+46.06988*(X)
37400 RETURN
37500 'prepare screen and axes for both types of graph
37600 CLS
37700 INPUT "Lower and Upper Bounds for X ? ",XLO,XHI
37800 IF XHI<XLO THEN GOTO 37700
37900 INPUT "Lower and Upper Bounds for Y ? ",YLO,YHI
38000 IF YHI<YLO THEN GOTO 37900
38100 XSCALE=220/(XHI-XLO) : YSCALE=150/(YHI-YLO)
38200 IF XHI<=0 THEN XORG%=270 ELSE IF XLO>=0 THEN XORG%=50
 ELSE XORG%=INT(50-XLO*XSCALE)
38300 IF YHI<=0 THEN YORG%=25 ELSE IF YLO>=0 THEN YORG%=175
 ELSE YORG%=INT(175+YLO*YSCALE)
38400 INPUT"Do you want X-axis labelled Y/N ? ",F$
38500 XLAB%=(F$="y") OR (F$="Y")
38600 INPUT"Do you want Y-axis labelled Y/N ? ",F$
38700 YLAB%=(F$="y") OR (F$="Y")
38800 PUT(0,0),SKREEN%,PSET
38900 LINE (XORG%,25)-(XORG%,175),15
39000 LINE (50,YORG%)-(270,YORG%),15
39100 XMARK=0 : MDIF=(XHI-XLO)/4
39200 FOR MARK%=1 TO 5
39300 PX%=50+INT(XMARK*XSCALE+.5)
39400 LINE(PX%,YORG%)-(PX%,YORG%+3),15
39500 IF NOT XLAB% THEN GOTO 40100
39600 NUM$=STR$(XMARK+XLO)
39700 IF XMARK+XLO>=0 THEN NUM$=RIGHT$(NUM$,LEN(NUM$)-1)
39800 IF LEN(NUM$)>9 THEN NUM$="..."
39900 X%=PX%-4*LEN(NUM$)+6 : Y%=YORG%+4
40000 GOSUB 13000 'narrow characteres
```

```
40100 XMARK=XMARK+MDIF


40200 NEXT MARK%
40300 YMARK=0 : MDIF=(YHI-YLO)/4
40400 FOR MARK%=1 TO 5
40500 PY%=175-INT(YMARK*YSCALE+.5)
40600 LINE(XORG%,PY%)-(XORG%-3,PY%),15
40700 IF NOT YLAB% THEN GOTO 41300
40800 NUM$=STR$(YMARK+YLO)
40900 IF YMARK+YLO>=0 THEN NUM$=RIGHT$(NUM$,LEN(NUM$)-1)
41000 IF LEN(NUM$)>9 THEN NUM$="..."
41100 X%=XORG%-4*LEN(NUM$)-1 : Y%=PY%-8
41200 GOSUB 13000 'narrow characteres
41300 YMARK=YMARK+MDIF
41400 NEXT MARK%
41500 RETURN
41600 'discrete graph
41700 GOSUB 45600
41800 IF DSK$="y" OR DSK$="Y" THEN GOTO 42500
41900 CLS: INPUT"Number of points ? ",NOP%
42000 DIM X(NOP%),Y(NOP%)
42100 FOR I%=1 TO NOP%
42200 PRINT"Type x("I%;") and y(";I%")";
42300 INPUT X(I%),Y(I%)
42400 NEXT I%
42500 FOR I%=1 TO NOP%-1
42600 FOR J%=I%+1 TO NOP%
42700 IF X(I%)>X(J%) THEN TEMP=X(I%) : X(I%)=X(J%) : X(J%)=TEMP :
 TEMP=Y(I%) : Y(I%)=Y(J%) : Y(J%)=TEMP
42800 NEXT J%
42900 NEXT I%
43000 GOSUB 37500 'prepare screen and axes
43100 PX%=INT(50+(X(1)-XLO)*XSCALE)
43200 P0Y%=PX%
43300 PY%=INT(175-(Y(1)-YLO)*YSCALE)
43400 PSET (PX%,PY%),0
43500 GOSUB 44800 'draw symbol
43600 FOR I%=2 TO NOP%
43700 PX%=INT(50+(X(I%)-XLO)*XSCALE)
43800 PY%=INT(175-(Y(I%)-YLO)*YSCALE)
43900 P1Y%=INT(50+(X(NOP%)-XLO)*XSCALE)
44000 P0Y%=INT(50+(X(1)-XLO)*XSCALE)
44100 IF NOT XLN% THEN GOTO 44300
44200 LINE -(PX%,PY%),15
44300 GOSUB 44800 'draw symbol
44400 NEXT I%
44500 'store picture
44600 GET (0,0)-(319,199),SKREEN%
44700 RETURN
44800 'draw symbol at (px%,py%)
44900 PSET (PX%+2,PY%+2),15
45000 LINE -(PX%-2,PY%+2),15
```

```
45100 LINE -(PX%-2,PY%-2),15
45200 LINE -(PX%+2,PY%-2),15
45300 LINE -(PX%+2,PY%+2),15
45400 PSET (PX%,PY%),15
45500 RETURN
45600 CLS:INPUT"Would you like line between points (Y/N) ? ",LN$
45700 XLN%=(LN$="Y") OR (LN$="y")
45800 INPUT "Data are in disk (Y/N) ? ",DSK$
45900 XDSK%=(DSK$="Y") OR (DSK$="y")
46000 IF NOT XDSK% THEN GOTO 47200
46100 INPUT"Name ? ",NA$
46200 DIM X(500),Y(500)
46300 OPEN "I",#1,NA$
46400 NOP%=0
46500 NOP%=NOP%+1
46600 INPUT#1,X(NOP%),Y(NOP%)
46700 PRINT X(NOP%),Y(NOP%)
46800 ON ERROR GOTO 47000
46900 GOTO 46500
47000 NOP%=NOP%-1
47100 CLOSE
47200 RETURN
47300 'switch color monitor
47400 DEF SEG=0 : KEY OFF
47500 IF MODE%=0 THEN INPUT"width",WYDTH% ELSE WYDTH%=40*MODE%
47600 POKE &H410,(PEEK(&H410) AND &HCF) OR &H10
47700 SCREEN 1,0,0,0 : SCREEN 0 : WIDTH WYDTH%
47800 LOCATE ,,1,6,7
47900 SCREEN MODE%,0
48000 IF MODE%=0 THEN COLOR FOREGROUND%,BACKGROUND%,BORDER%
 ELSE IF MODE%=1 THEN COLOR BACKGROUND%,PALETTE%
48100 IF MODE<>0 THEN LINE (0,0)-(320*MODE%-1,199),PAPER%,BF
48200 RETURN
48300 RETURN
```

This is a simple program which runs on an IBM - PC or other microcomputer which runs suitable BASIC software. Figure APF1 shows how to use it and figure APF3 shows the lists of it.

Figure APF1 - Use of the PLOTTER Program.

Press F2, then

```
 Simulation (Y/N) ? N
 Axis names (Y/N) ? Y
Lower and Upper bounds for X ? 0,60
Lower and Upper bounds for Y ? 0,6
Label - X - ? Biomass (g/l)
Label - Y - ? Time (hours)
Which file ? b:wt301.dat
Which colour (1-black,2-blue,3-red,4-green) ? 1
Which symbol ? 15
Which line (0-8) ? 0
Which Label position (0-8)? 1
Label ? Wild type
```

Following the instructions described in figure APF1, we can get the graph shown in figure APF2.

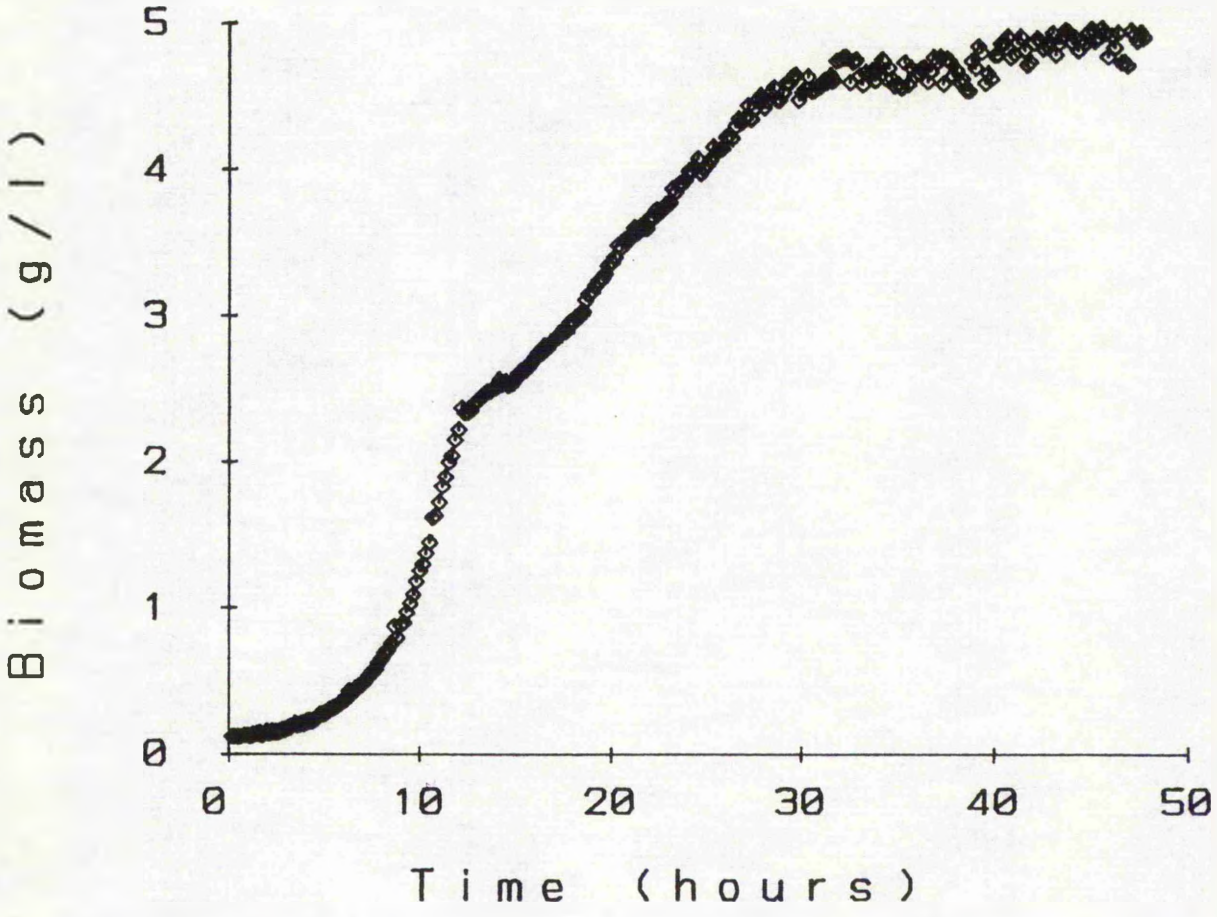


Figure APF2 - Growth curve of the Saccharomyces cerevisiae Σ1278 wild type in batch culture.

Figure APF3 - Program PLOTTER.

```
100 REM ***** APPENDIX E - PLOTTER - Epson HI - 80 *****
200 CLS
300 LPRINT"vs1"
400 LPRINT"sp4"
500 DIM X(1000),Y(1000),Y1(1000)
600 INPUT " Simulation (Y/N)";SI$
700 INPUT " Axis names (Y/N)";AX$
800 INPUT "Lower and Upper bounds for x?",XLO,XHI
900 INPUT "Lower and Upper bounds for y?",YLO,YHI
1000 XSCALE=1300/(XHI-XLO):YSCALE=140/(YHI-YLO)
1100 LPRINT "ma 0,0"
1200 IF AX$="Y" THEN 1600
1300 IF AX$="N" THEN 5200
1400 IF AX$<>"Y" THEN  700
1500 IF AX$<>"N" THEN 700
1600 FOR I=1 TO 3
1700 LPRINT "ma 200,350": LPRINT "di 0,90": LPRINT "ax 1,200,5"
1800 NEXT I
1900 J=0
2000 I2=(YHI-YLO)/5:I1=0
2100 FOR I=0 TO 5
2200 I1=I1+I2
2300 LPRINT "ma";215+J;",";200: J=J+200 : LPRINT "si 34,36":
 LPRINT "Em1"
2400 LPRINT "la";ABS(YHI-(I1-I2))
2500 NEXT
2600 INPUT "Label - Y - ";IN$:X3=LEN(IN$):X4=X3:X3=X3*50
2700 LPRINT "di -360,0"
2800 K=0
2900 FOR J=1 TO X4
3000 FOR I=1 TO 1
3100 LPRINT "ma";700-(X3/2)+K-I;",";100-I
3200 LPRINT "si";40+(I*3);",";(40+(I*3))
3300 LPRINT "la";MID$(IN$,ABS(X4+1-J),1)
3400 NEXT I
3500 K=K+60
3600 NEXT J
3700 FOR I=1 TO 3
3800 LPRINT "ma 1200,350": LPRINT "ax 2,1300,5"
3900 NEXT I
4000 LPRINT"di 0,90"
4100 J=0
4200 I3=(XHI-XLO)/5:I4=0
4300 FOR I=0 TO 5
4400 I4=I4+I3
4500 LPRINT "ma";1280;","280+J: J=J+260:LPRINT "si 34,36":
 LPRINT "em1":LPRINT "la";(INT((I4-I3)*100))/100
4600 NEXT
4700 INPUT "Label - X - ";L$:X5=LEN(L$):X6=X5:X5=X5*50
4800 LPRINT "di 0,90"
4900 K=0:FOR J=1 TO X6:FOR I=1 TO 1
5000 LPRINT "ma";1400-I;",";(900-(X5/2))+K-I:
 LPRINT "si";40+(I);",";40+(I)
```

```
5100 LPRINT "la";MID$(L$,J,1):NEXT I:K=K+60:NEXT J
5200 F=0
5300 F=F+1
5400 LPRINT "si 80,80"
5500 INPUT"Which is file";FI$(F)
5600 OPEN "i",#1,FI$(F)
5700 T=L=0
5800 L=L+1
5900 ON ERROR GOTO 6400
6000 INPUT #1,X(L),Y1(L)
6100 Y(L)=(Y1(L))
6200 GOTO 5800
6300 CLOSE
6400 T=0
6500 FOR I=1 TO L
6600 T=T+1
6700 X(T)=X(I): Y(T)=Y(I)
6800 NEXT I
6900 INPUT"Which colour (1-black,2-blue,3-red,4-green)";CL:
 LPRINT "sp";CL
7000 INPUT"Which is symbol";SYS:
 INPUT "Which is line Type 0-8";TY
7100 LPRINT"em1"
7200 PX%=INT(350+((X(1)*1300)/(XHI-XLO))):
 PY%=INT(1200-((Y(1)*1000)/(YHI-YLO)))
7300 P0Y%=PX%
7400 P1%=INT(350+((X(T-1)*1300)/(XHI-XLO)))
7500 GOSUB 8600
7600 FOR I=2 TO T-1
7700 PX%=INT(350+((X(I)*1300)/(XHI-XLO)))
7800 PY%=INT(1200-((Y(I)*1000)/(YHI-YLO)))
7900 PRINT PX%,PY%
8000 GOSUB 8600
8100 NEXT
8200 IF SI$="Y" THEN 8900
8300 IF SI$="N" THEN 10600
8400 IF SI$<>"Y" THEN 10600
8500 IF SI$="N" THEN 10500
8600 REM point
8700 LPRINT "ma";PY%;",";PX%:LPRINT "si 20,20":LPRINT"am";SYS
8800 RETURN
8900 FOR H=0 TO 1
9000 X=(P0Y%-350)/XSCALE+XLO:GOSUB 10300
9100 PY%=INT(1200-((Y*1000)/(YHI-YLO)))
9200 LPRINT "ma";PY%",";P0Y%
9300 FOR PX%=350 TO 2000  STEP 20
9400 X=(PX%-350)/XSCALE+XLO:GOSUB 10300
9500 PY%=INT(1200-((Y*1000)/(YHI-YLO)))
9600 IF PX%<P0Y% THEN GOTO 9900
9700 IF PX%>P1% THEN GOTO 10100
9800 LPRINT "lt";TY:LPRINT "da ";PY%;",";PX%+H
9900 NEXT PX%
10000 GOTO 10500
```

```
10100 NEXT H
10200 GOTO 10500
10300 Y=3.114-.060969*(X)
10400 RETURN
10500 LPRINT "ma 0,0"
10600 REM label
10700 INPUT "Which is it ";F
10800 PRINT "label ";F
10900 INPUT "label";LB$
11000 X5=LEN(LB$):X6=X5:X5=X5*50
11100 LPRINT "di 0,90"
11200 LPRINT "ma 1700,300"
11300 LPRINT "ma";1480+(F*80);",";300:LPRINT "si 30,30":LPRINT "am";SYS
11400 K=0:FOR J=1 TO X6
11500 LPRINT "ma";1500+(F*80);",";350+K:LPRINT "si 30,30"
11600 LPRINT "la";MID$(LB$,J,1):K=K+30:NEXT J
11700 END
```

8. - REFERENCES.

1. ----- (1987). The Economist, Jan., 17, 23-26.

2. Anda, E.V. and Iglesias, J.R. (1985). Rev. Bras. Tecnol. 6, 34-39.

3. House, R. (1986). South., Sept., 42.

4. Barros, P.M. (1985). Rev. Bras. Tecnol., 2, 5-22.

5. Joyce, C. (1985). New Scientist, 18, 36-41.

6. Dahlman, C. and Westphal, L. (1983). Finance & Development, Dec., 6-9.

7. Goodrich, R. (1982). In:Commercial Biotechnology: An Introduction Analysis Congress of the United States., Washington D.C..

8. Carioca, J.O.B., Arora, H.L., and Khan, A.S. (1981). In: Advances in Biochemical Engineering, (Fiechter, A., Ed.), Springer-Verlag, Berlin GDR., 20, 153-161.

9. House, R. (1986). South, July, 99.

10. ----- (1985). In:The Economist, 9, Feb., 49.

11. House, R. (1986). South, Nov., 107-108.

12. Hordeski, M.F. (1984). In: Microprocessors in Industry, Van Nostrand Reinhold Company, California. 1-149.

13. Wang N.S. and Stephanopoulos G.N. (1984). In: CRC Critical Reviews in Biotechnology,(G.G. Stewart and I. Russel, eds.), CRC PRESS, London, 2, 1-103.

14. Rolf, M.J. and Lim H.C. (1982). Enzyme Microb. Technol., Vol. 4, 11, 370-380.

15. Fuld, G.J. (1960).Advan. Appl. Microbiol., 2, 351.

16. Argimer W.B. and Hamprey A.E. (1979). In: Microbial Technology, (Peppler H.L. and Perlman D., 2nd ed.), Acad. Press, London, 2, 375-397.

17. Nyiri, L.K (1972). In: Advances in Biochemical Engineering, (T.K. Ghose, A. Fiechter and N. Blakebrough), Springer-Verlag, New York, 2, 49-93.

18. Dobry, D.D. and Jost, J.L. (1979). Biotechnol. Bioeng. Symp., 9, 39-47.

19. Wiegand, W.A. (1978).In:Annual Reports on Fermentation Processes, (Peppler, H.J., Ed.), Acad. Press, New York,2, 43-72.

20. Zabriskie, D.W., Arminger, W.B. and Humprey, A.E. (1977). In:Workshop Computer Applications in Fermentation Technology, Verlag Chemie, New York, 3, 59.

21. Hampel, W.A. (1979). In:Advances in Biochemical

engineering, (Ghose, T.K., Fiechter, A., and Blakebrough, N., Eds.), Springer-Verlag, Berlin, 13, 1.

22. Hatch, R.T. (1982). In: Annual Reports on Fermentation Processes. (Tsao, G.T., ed.) Academic Press, New York, 5, 291-311.

23. Buckland, B.C. (1984). Bio/technology, 27, 756-760.

24. Merrill, R.D. and Bauer, K. (1986). Biotechnol. Bioeng., 28,494-503.

25. Jefferies, R.P.III, Klein, S.S., and Drakeford, J. (1979). Biotechnol. Bioeng. Symp., 9, 231-239.

26. Forrest, E.H., Jansen, N.B., Flickinger, M.C. and Tsao, G.T. (1981). Biotechnol. Bioeng., 23, 455-460.

27. Yano, T., Kobayashi, T. and Shimizu. (1981). J. Ferment. Technol., 59, 295-301.

28. Pungor, E.Jr., Schaefer, E.J., Cooney, C.L. and Weaver, J.C. (1983). Eur. J. Microbiol. Biotechnol., 18, 135-140.

29. Bayer, K. (1984). Acta Biotechnol., 4, 209-234.

30. Mathers, J. and Cork, D.J. (1984). Biotechnol. Lett., 6, 87-90.

31. Shimizu, N., Sumitani, T. and Odawara, Y. (1984). Biotechnol. Bioeng. Symp., 27, 192-201.

32. Titus, J.A., Luli, G.W., Dekleva, M.L. and Strohl, W.R. (1984). Appl. Environ. Microbiol., 47, 239-244.

33. Podruzny, M.F. and Van den Berg, L. (1984). Biotechnol. Bioeng., 26, 392-393.

34. Horan, N.J., Eccles, C.R. and Mara, D.D. (1985). Process Biochem., 20,13-16

35. Dinwoodie, R.C. and Mehnert, D.W. (1985). Biotechnol. Bioeng., 27, 1060-1062.

36. Strohl, W.R., Schlasner, S.M. and Lorensen, P.L. (1986). BioTechiques, 4, 336-344.

37. Chen, K.C., Wu, W.T and Chang, K.Y. (1986). Appl. Microbiol. Biotechnol., 24, 113-116.

38. Yasuda,T. (1981). In:3th Int. Conf. on Computer Appl. in Ferment. Technol., Manchester.

39. Nanba, A., Hirota, F. and Nagai, S.J. (1981). J. Ferment. Technol., 59, 383-389.

40. Wang, H.Y., Cooney, C.L. and Wang, D.I.C. (1977). Biotechnol. Bioeng. 19, 69-86.

41. Luttman, R., Bellgardt, K.H., Bartzke, U., Nothnagel, J., Rapmundt, W. and Wonka, F. (1984). 3th Eur. Congress on Biotechnol., 585-592.

42. Lundell, R. (1977). Biotechnol. Bioeng., 9, 381-393.

43. Beaverstock, M.C. and Trearchis, G.P. (1979). Biotechnol. Bioeng. Symp., 9, 241.

44. Bristol, E.H. (1977). Control Eng., 21, 24-26.

45. Spector, A.Z. (1984), Scientific American, 251, 126-139.

46. Kay, A. (1984), Scientific American, 251, 40-47.

47. Hennigan, P.J. and Lim, H.C. (1981).Annual Winter AIChE Meeting, Orlando, feb.

48. Meiners, M. and Rapmundt, W. (1983). Biotechnol. Bioeng., 5, 809-844.

49. Hsin, T.C. (1983). INFO, 5 1-8.

50. Nawa, O. (1983). INFO,4 Supplement.

51. Humphrey, A.E. (1971). In:Proc. of LABEX Symp. on Computer Control of Fermentation Processes., London, 1.

52. Onken, V. and Wieland, P. (1985). In:Biotechnology, (Brauer, H. ed.), vol 2., VCH, Speyer West Germany, 787-806.

53. Waite, P. and Gray, P.P. (1977).Biotechnol. Bioeng., 19, 575-581.

54. Mou, D.G. and Cooney, C.L. (1983). Biotechnol. Bioeng., 25, 225.

55. Camberbach, D.M. and Bu'lock, J.D. (1983). Biotechnol. Bioeng., 25, 2503.

56. Märkl, H., Mather, M. and Witty,W. (1984).In:Proc. 3rd European Congress on Biotechnology, 3, 137.

57. Wu, T.W., Chen K.C. and Chiou, H.W. (1984). Biotechnol. Bioeng., 27, 756-760.

58. Rolf, M.J. and Lim H.C. (1985). Biotechnol. Bioeng., 27, 1236-1245.

59. Miskiewicz, T. and Miszczak, W. (1985). Biotechnol. Lett., 7, 365- 368.

60. Walczak, P. (1986).In:Proc. 14th Int. Con. of Microbiol.. Manchester, UK, 87.

61. Fredrickson, A.G. Megee, R.D.,III, and Tsuchiya. (1970).Adv. Appl. Microbiol., 13, 419

62. Roels, J.A. (1980). Biotechnol. Bioeng., 22, 2457.

63. Yamashita, S., Hoshi, H., Inagaki, T. (1969). In:Fermentation Advances. (Perlman, D., Ed.), Acad. Press, 441.

64. Cooney, C.L., Wang, H.Y. and Wang, D.I.C. (1977). Biotechnol. Bioeng., 19, 55.

65. Zabriskie, D.W., Arminger, W.B. and Humprey, A.E. (1977). In:Workshop Computer Applications in Fermentation Technology, 3, Verlag Chemie, New York, 59.

66. Monod, J. (1949). Ann. Rev. Microbiol., 3, 371-94.

67. Leudeking, R., Pirt , E. L. (1959). J. Biochem. Microbiol. Technol. Eng., 1, 393.

68. Bolle, W.L, Van Breugel, J., Van Eybergen, G.C., Kossen, N.W.F. and Van Gils, W. (1986). Biotechnol. Bioeng., 28, 1621-1636.

69. Monod, J. (1942).In:Recherches sur La Croissance des Cultures Bacteriennes. 2nd., Hermann, Paris.

70. Fredrickson, A.G. and Tsuchiya, H.M. (1977). In:Chemical Reactor Theory, (L. Lapidus and N.R. Amundson, Eds.), Prentice-Hall, Englewood Cliffs, N.J., 405.

71. Garfinkel, D. (1981). TIBS, Mar. 69-71.

72. Kossen, N.W.F.(1982).In: 3rd Int. Conf. on Computer Applications in Fermentation Technology., London, 23.

73. Takamatsu, T., Shioya, S., Yokoyama, K. and Ihara, D. (1979). Biotechnol. Bioeng. Symp. 9, 283.

74. Boutlon, R.(1979). Biotechnol. Bioeng. Symp. 9, 167.

75. Barford, J.P. (1981). Biotechnol. Bioeng., 23, 1735-1762.

76. Barford, J.P. and Hall, R.J. (1981).Biotechnol. Bioeng., 23, 1735-1762.

77. O'Neil, D.G. and Lyberatos, G.(1986).Biotechnol. Bioeng., 28, 1323-1333.

78. Lim, H.C., Tayeb, Y.J., Modak, J.M. and Bonte,P. (1986). Biotechnol. Bioeng., 28, 1408-1420.

79. Dhurjati, P., Ramkrishna, D., Flickinger, M.C. and Tsao, G.T. (1985). Biotechnol. Bioeng., 27, 1-9.

80. Rolf, M.J. and Lim, H.C. (1985). Biotechnol. Bioeng., 27, 1236-1245.

81. Pontryagen, L.S., Bottyanski, V.G., Gamkrelidge, R.V. and Mishenko, E.F. (1962). In:The Mathematical Theory of Optimal Processes., Wiley, New York

82. Fox, M.J. (1965). Comput. J., 8, 42.

83. Aiba, S. (1979). Biotechnol. Bioeng. Symp. 9, 269-281.

84. Zabriskie, D.W. (1979). Annals New York Academy of Sciences., 326, 223.

85. Wiegand, W.A. (1978).In:Annual Reports on Fermentation Processes, 2, (Peppler, H.J., Ed.), Acad. Press, New York, 43-72.

86. Bungay, H.R. (1984). Biotechnol. Bioeng. Symp., 14, 611.

87. Raviv, R. and Ben-Yaakov, S. (1984). Biotechnol. Bioeng., 26, 1239-1244.

88. Shoda, M. and Ishikawa, Y. (1981). Biotechnol. Bioeng., 23, 801.

89. King, J.D. and White, D.C. (1977). Appl. Env. Microbiol., 33, 777-783.

90. Harris, C.M. and Kell, D.B. (1985). Biosensors, 1, 17-84.

91. Karl, D.M. (1980). Microbiol. Rev., 44, 739-796.

92. Griffiths, M.W. and Phillips, J.D. (1986). Proceedings of XIV Int. Cong. of Microb., Manchester,U.K., 76.

93. Allen, P.D. (1972). Dev. Ind. Microb., 14, 67-73.

94. Thore, A., Asehn, A., Lundin, A. & Bergman, S. (1975). J. Clin. Microbiol., 1, 1-8.

95. Lovgren, T. Peacock, R., Lavi, J., Karp, M. & Raunio, R. (1982). Int. Lab., 12, 58-67.

96. Zabriske, D.W. (1979). Biotechnol. Bioeng. Symp., 9, 117.

97. Sakato, K., Tanaka, H. and Samejima, H. (1981). Proc. N. Y. Acad. Sci., 369, 321.

98. Harrison, D.E.F. and Chance, B. (1970). J. Appl. Microbiol., 19, 446.

99. Matsunaga, T., Karube, I & Suzuki, S. (1979). Appl. Env. Microbiol., 37, 117-21.

100. Blachere, H.T. and Peringer, P. (1978). In:1st European Congr. Biotechnology. Verlag Chemie, New York, 65.

101. Cheruy, A. and Durand, A. (1979). Biotechnol. Bioeng. Symp., 9, 303.

102. White, D.C. (1980). Abstr. 2nd Int. Symp. Microbial Ecology, Warwick, 77.

103. Newman, J. and O'Brien, R.T. (1975). Appl. Env. Microbiol., 30, 584-588.

104. Warren, L.S., Benoit, R.E. and Jessee, J.A. (1978).Appl. Env. Microbiol., 35, 136-141.

105. Trinel, P.A., Hanoune, A. and Leclerc, H. (1980). Appl. Env. Microbiol., 39, 976-982.

106. Levin,G.V. (1963). Adv. Appl. Microbiol., 5, 95-133.

107. Humphrey, A.E. (1974). Chem. Eng, 8, 98-112.

108. Norris, J.R. and Swain, H. (1981). In:Methods in Microbiology, Vol. 5A (J.R. Norris and D.W. Ribbons, Eds.), Acad. Press, London, 105-34.

109. Adams, R.P.L. (1980). In:Laboratory Techniques in Biochemistry and Molecular Biology. (T.S. Work and R.H. Burdon, Eds.), 8, Elsevier, Amsterdam, 103-108.

110. Jones, J.G. (1979). A Guide to methods for Estimating Microbial Numbers and Biomass in Fresh Water, Freshwater Biol. Assoc., Scientific Publication, 39.

111. Koch, A.L. (1961). Biochim. Biophys. Acta., 51, 241-249.

122. Powell, E.O. and Stoward, P.J. (1962). J. Gen. Microbol., 27 489-500

113. Wilson, M.C. and Harvey, J.D. (1983). Biophys. J., 41, 13-21.

114. Lloyd, D. and Scott, R.I. (1983). J. Microbiol. Methods., 1, 313-328.

115. Gates, M.A. (1983). J. Gen. Microbiol., 129, 895-900.

116. Hancher, C.W., Thacker, L.H. and Phares, E.F. (1974). Biotechnol. Bioeng., 16, 475-484.

117. Aldridge, C., Jones, P.W., Gibson, S., Lanham, J., Meyer, M., Vannest, R. and Charles, R. (1977). J. Clin. Microbiol., 6, 406-413.

118. Heslop, J. (1980). Clin. Lab. Products, Nov. issue, 1-10.

119. Gibson, S.(1982). In:Proc. 3rd. Int. Symp. Rapid Methods and Automation in Microbiology, Washington, 80-82.

120. Ohashi, M., Watabe, T., and Watanabe, Y. (1979). Biotechnol. Bioeng. Symp., 9, 103-116.

121. Pirt, S.J. and Kurowski, W.M. (1970). J. Gen. Microbiol., 63, 375-366.

122. Kull, F.C. and Cuatrecasas, P. (1986). Appl. Biochem. Biotechnol., 8, 97-103.

123. Nestaes, E. and Wang, D.I.C. (1981). Biotechnol. Bioeng., 23, 2803-2813.

124. Nestaes, E. and Wang, D.I.C. (1983). Biotechnol. Bioeng., 25, 1981-1987.

125. Stewart, G.N. (1899). J. Exp. Med., 4, 235-243.

126. Anon. (1981). Detection Times (Bactomatic Inc.), 1.

127. Grover, N.B., Naaman, J., Ben-Sasson, S. and Doljaniski, F. (1969). Biophys. J., 9, 1415-1425.

128. Grover, N.B., Ben-Sasson, S.A. and Naaman, J. (1982). In: Cell Analysis, 1 (N. Catsimpoolas, Ed.), Plenum Press, New York, 93-109.

129. Wilkins, J.R. (1974). Appl. Microbiol., 27, 135-140.

130. Wilkins, J.R. (1979). Appl. Env. Microbiol., 36, 683-687.

131. Gualandi G. (1973). Biotechnol. Bioeng. Symp., 4, 799.

132. Guilbault, G.C. (1981). Annals New York Academy of Science, 369, 285.

133. Ianniello, R.M. and Jespersen, N.D. (1982). In: Comprehensive Analytical Chemistry, 12B, Elsevier, New York, chap. 5.

134. Suzuki, S. and Karube, I. (1981). In:Advances in

Biotechnology, 3, Pergamon Press, Toronto, 355.

135. Forest, W.W. (1972). In:Methods in Microbiology, 6B (R. Norris and D.W. Ribbons, Eds.), Acad. Press, London, 285-318.

136. Pungor, E., Schaeter, E., Weaver, J. C. and Cooney, C.L. (1981). In: Advances in Biotechnology,(Moo-Young,M., ed.), Pergamon Press, Toronto, 1, 393-398.

137. Tonge, G.M. (1981). 3rd Int. Conf. on Computer Appl. in Ferment. Technol., Manchester.

138. Buckland, B.C. (1981). 3rd Int. Conf. on Computer Appl. in Ferment. Technol., Manchester.

139. Mor, J.R. (1976). In:Proc. 2nd Int. Symp. Rapid Methods and Automation in Microbiology, (H.H. Johnston and S.W.B. Newsom, Eds.), Learned Information (Europe) Ltd, Oxford, 7.

140. Calvet, E. and Prat, H. (1963). In:Recent Progress in Microcalorimetry, MacMillan, New York, 1980.

141. Comberbach, D.M. and Bu'lock, J.D. (278).Biotechnol. Bioeng., 25, 2503.

142. Bach, H.P., Woehrer, W. and Roehr, M. (1978). Biotechnol. Bioeng., 20, 799.

143. Hatch, R.T., Wilder, C., and Cadman, T.W. (1979). Biotechnol. Bioeng. Symp., 9, 25.

144. Fazel-Madjlessi, J. and Bailey, J.E. (1979). Biotechnol. Bioeng., 21, 1995.

145. Ramsay, G., Franklin A., Steel, D., Cardosi, M., Shneider, B., Finnegan, R. and Turner, A.P.F. (1986). In:Proc. 14th Int. Con. of Microbiol., Manchester, U.K., 80.

146. Jackman, P.J.H. and Fernandes, M.A. (1986). In:Proc. 14th Int. Cong. of Microbiol., Manchester, U.K.,79.

147. Stewart, G.G., Bilinski, C.A., Russell, I., and Sills, A.M. (1985). In:Microbiology - 1985, (Leive, L., Ed.), Washington, D.C., 367-374.

148. Harder, W., and Dijkhuizen, L.(1983). Ann. Rev. Microbiol., 37, 1-23.

149. Nagasu, T., and Hall, B. D. (1985). Gene, 37, 247-253.

150. Moye, S. W., Amuro, N., Rao, J. K. M., and Zalkin, H. (1985). J. Biol. Chem., 14, 8502-8508.

151. Cooper, T.G. (1982). In:The Molecular Biology of the Yeast Saccharomyces: Metabolism and Gene Expression. (Strathern, J. N., Jones, E. W., and Broach, J.R., Eds.), New York, Cold Spring Harbor Laboratory. 39.

152. Vichido, I., Mora, Y., Quint, C., Palacios, R., and Mora, J. (1978). J. Gen. Microbiol., 106, 251-259.

153. Bogonez, E., Satrústegui, J., and Machado, A. (1985). J. Gen. Microbiol., 131, 1425-1432.

154. Magasaniz, B., Prival, M. J., Brenchley, J. E., Tyler, B. M., De Leo, A. B., Streicher, S. L., Bender, R. A., and Paris, C. G. (1974). Current Topics in Cellular Regulation., 8, 119-138.

155. Meers, J. L., and Tempest, D. W. (1970). J. Gen. Microbiol., 64, 187-194.

156. Tempest, D. W., Meers, J. L., and Brown, C. M. (1970). Biochemical J., 117, 405-407.

157. Barratt, R. W. (1963). J. Gen. Microbiol., 33, 33-42.

158. Burn, V. J., Turner, P. R., and Brown, C. M. (1974). Antonie van Leeuwenhoek, 40, 93-102.

159. Tempest, D. W., Meers, J. L., and Brown, C. M. (1973). In:The enzymes of Glutamine Metabolism., (Prusiner, S. and Stadtman, E. R., Eds.), New York, Academic Press., 167-182.

160. Roon, R. J., Even, H. L., and Larimore, F. (1974). J. Bacteriology, 118, 89-95.

161. Legrain, C., Vissers, S., Dubois, E., Legrain, M., and Wiame, J. M. (1982). Eur. J. Biochem., 123, 611-616.

162. Brown, C. M. (1976). In: Continuous Culture 6: Applications and New Fields. (Dean, A.C.R., Ellwood, D.C., Evans, C.G. and Melling, J., Eds.), London, Ellis Horwood LTD., 170-180.

163. Doherty, D.(1970). In:Methods in Enzymology, 17, New York, Academic Press, 850-856.

164. De Castro, I.N., Ugarte, M., Cano, A., and Mayor, F. (1970). Eur. J. Biochem., 16, 567-570.

165. Roon, R. J., and Even, H. L. (1973). J. Bacteriol., 116, 367-372.

166. Kohlhaw, G., Gragert, W., and Holzer, H. (1965). Biochem. Z., 341, 224.

167. Uno, I., Matsumoto, K., Adachi, K., and Ishikawa, T. (1984). J. Biol. Chem., 259, 1288-1293.

168. Mazon, M. J., and Hemmings, B. A. (1979). J. Bacteriol., 139, 686-689.

169. Hernandez, G., Sánchez-Pescador, R., Palacios, R. and Mora, J. (1983).J. Bacteriol., 154, 524-528.

170. Gonzalez, Alicia, Rodriguez, L., Olivera, H., and Soberon, M. (1985). J. Gen. Microbiol., 131, 2565-2571.

171. Bogonez, E., Satrústegui, J., and Machado, A. (1983). Biochim. Biophys. Acta, 733, 234-241.

172. Roon, R. T., Even, H. L., Dunlop, P., and Larimore, F. L. (1975). J. Bacteriol., 122, 502-509.

173. Botstein, D., Davis, R. W. (1982). In:The Molecular Biology of the Yeast Saccharomyces: Metabolism and Gene Expression. (Strathern, J. N., Jones, E. W., and Broach, J.R., Eds.), New York, Cold Spring Harbor Laboratory. 607.

174. Gunge, N. (1983). Ann. Rev. Microbiol., 37, 253-276.

175. Lauer, G. D., Roberts, T. M., Klotz, L. C. (1977). J. Mol. Chem., 114, 507-526.

176. Kohli, J., Hottinger, H., Munz, P., Strauss, A., Thuriaux, P. (1977). Genetics, 87, 471-489.

177. Ephrussi, B., de Margerie-Hottinguer, H., Roman, H. (1955). Proc. Natl. Acad. Sci. USA., 41, 1065-1071.

178. Bevan, E. A., Mitchell, D. J. (1979). In:Virus and Plasmid in Fungi, (Lemke, P.A., Ed.), Basel, Dekker, 191-199.

179. Gubbins, E. J., Newlon, C. S., Kann, M. D., Donelson, J. E. (1977).Gene, 1, 185-207.

180. Zakian, V. A., Brewer, B. J., Fangman, W. L. (1979). Cell, 17, 923-934.

181. Gerbaud, C., Guerineau, M. (1980). Curr. Genet., 1, 219-228.

182. Futcher, A.B., and Cox, B. S. (1983). J. Bacteriol., 154, 612-622.

183. Old and Primrose (1985).In:Principles of Gene Manipulation, Blackwell Scientific Publications, London.

184. Imanaka and Aiba (1981). Annals New York Academy of Sciences., 369, 1-14.

185. Caulcott, C. A. (1984). Biochemical Society Transactions, 1140-1141.

186. Srienc, F., Campbell, J. L., and Bailey, J. E. (1986). Biotechnol. Bioeng., 28, 996-1006.

187. Walmsley, R. M., Gardner, D. C. J., and Oliver, S. G. (1983). Mol. Gen. Genet, 192, 361-365.

189. San, K. and Stephanopoulos, G. (1984).Biotechnol.

Bioeng., 26, 1209-1218.

190. Ackland, M.R., Manvell, P.M. and Bean, P.G. (1984). Biotechnol. Lett., 6, 137-142.

191. Bakers, A.S. and Somers, P.J. (1978). In:Topics in Enzyme and Fermentation Biotechnology, 2, Ellis Horwood, Sussex, England, chap. 3.

192. Cheung, T.D., Chan, W.K. and Kouvel, J.S. (1982). Rev. Sci. Instrum., 53, 701.

193. Cashmore, A. M., Albury, M. S., Hadfield, C., and Meacock, P.A. (1986). Mol. Gen. Genet., 203, 154-162.

194. Chevallier, M. R., and Aigle, M. (1979). FEBS Lett., 1, 179-180.

195. Pirt, S. J. (1975). In: Principles of Microbe and Cell Cultivation, Blackwell Scientific Publications, London.

196. Primrose, S. B., Derbyshire, P., Jones, I. M., Robinson, A., and Ellwood, D. C. (1984). In: Continuous Culture 8: Biotechnology, Medicine and the Environment., (Dean, A.C.R., Ellwood, D.C., and Evans, C.G.T., Eds.), Ellis Horwood Ltd., Chichester, 169-184.

197. Matin, A. (1981). In: Continuous Culture of Cells, (Calcott, P.H., ed.), CRC Press, New York, 69-97.

198. Calcott, P.H. (1981): In: Continuous Culture of Cells, (Calcott, P.H., ed.), CRC Press, New York, 127-140.

199. Tempest, D. W. (1970). Adv. Microb. Physiol., 4, 223.

200. Dykhuizen, D. E., and Hartl, D. L. (1983). Microbiol. Rev., 2, 150-168.

201. Bernfield, P. (1955). In: Methods in Enzymology. (Collowick, S. P. and Kaplan, N. O., Eds.), Vol. 1, 149-150. Academic Press, New York.

202. Llayne, E. (1957). In: Methods in Enzymology. (Collowick, S. P. and Kaplan, N. O., Eds.), Vol. 3, 450-451. Academic Press, New York.

203. Bradford, M. M. (1976). Analytical Biochemistry, 72, 248-254.

204. Perret, C. J. (1954). Nature, 27, 1012-1013.

205. Sargent, M. G. (1968). J. Bacteriol., 95, 1493-1494.

206. Collins, J. F., and Sherratt, D. J. (1973). J. Gen. Microbiol., 76, 217-230.

207. Aiba, S., and Koizumi, J. (1984). Biotechnol. Bioeeng., 26, 1026-1031.

208. Imanaka, T., Tanaka, T., Tsunekawa, H. and Aiba, S. (1981). J. Bacteriol., 147, 776-786.

209. Bergmeyer, H. U. (1974). Methods in Enzymatic Analysis, 2nd ed., Vols. III and IV, Academic Press, New York.

210. -------- (1980).Z80-Assembley Language Programming Manual. Zilog, California, USA.

211. ----- (1984). Databook Z80 Microprocessor Family. 4th. Ed., SGS-ATES, Italy.

212. ----- (1984). Technical Manual Z80 Microprocessor Family. 2nd. Ed., SGS-ATES, Italy.

213. ----- (1981). CMOS Databook, National Semiconductor Corporation, California, U.S.A.

214. ----- (1985). Microsystem Components Handbook Vol. 2, Intel, California, U.S.A.

215. Angell, I. O. (1985). In: Advanced Graphics with the IBM.. MacMillan Publishers LTD, London.

216. Leventhal, L. A. (1979). In:Z80 Assembley Language Programming by Lance A. Leventhal. Osborne McGraw-Hill, California, USA.

217. Topiwala H. H., and Hamer, G. (1971). Biotechnol.

Bioeng., 13, 919.

218. Lee, G. M., Song, K. B., Rhee, S. K., and Han, M. H. (1986). Biotechnol. Lett., 6, 385-390.

219. Helling, R. B., Kinney, T. and Adams, J. (1981). J. Gen. Microbiol., 123, 129-141.

220. Grenson, M., Dubois, E. and Piotrowska, M. (1974). Molec. Gen. Genet., 128, 73-85.

221. San, K. and Stephanopoulos, G. (1984).Biotchnol. Bioeng., 26, 1209-1218.

222. Wiame, J.M, Grenson, M. and Arst JR., H. N. (1985). Advances in Microbial Physiology, 26, 1-88.

223. Pamment, N. B., Hall, R. J. and Barford, J. P. (1978). Biotechnol. Bioeng., 20, 349-381.

224. Suomalainen, H. and Oura, E. (1971). In: The Yeasts. (Rose, A. H. and Harrison, J. S, Eds.), vol. 2, Acadamic Press, London, 3-60.

225. Roon, R. J., Even, H. L., Dunlop, P. and Larimore, F. L. (1975). J. Bacteriol., 121, 502-509.

226. Stadtman, E. R., and Ginsburg, A. (1974). In: The Enzymes, (Boyer, P.D., Ed.), 3rd Ed., Vol.10, Academic Press, New York, 755-807.

227. Bogonez, E., Satrústegui, J. and Machado, A. (1979). FEMS Microbiol. Lett., 6, 151-154.

228. Bossinger, J., Lawther, R. P. and Cooper, T. G. (1974). J. Bacteriol., 3, 821-829.

229. Kleinman, M. J., Gingold, E. B. and Stanbury, P. F. (1986). Biotechnol. Lett., 4, 225-230.

230. Koizumi, J, Aiba, S. (1986). Biotechnol. Bioeng., 28, 311-313.

231. Mead, D. J., Gardner, C. J. D. and Oliver, S. G. (1986). Biotechnol. Lett., 6, 391-396.

232. Futcher, A. B. and Cox, B. S. (1984). J. Bacteriol., 157, 283-290.

233. Petes, T. D. and Williamson, D. H. (1975). Cell, 4, 249-253.

234. Jayaram, M., Li, Y. Y., Broach, J. R. (1983). Cell, 34, 95-104.

235. Grayson, P. (1969). Process Biochem., 4, 43-61.